

ივ. ჯავახიშვილის სახელობის თბილისის სახელმწიფო უნივერსიტეტი
ზუსტ და საბუნებისმეტყველო მეცნიერებათა ფაკულტეტი

ნინო შენგელია

მოთხოვნების დამუშავება და ოპტიმიზაცია

სამაგისტრო პროგრამა: ინფორმაციული ტექნოლოგიები

სამაგისტრო ნაშრომი შესრულებულია ინფორმაციული ტექნოლოგიების
მაგისტრის აკადემიური ხარისხის მოსაპოვებლად

ხელმძღვანელი: მანანა ხაჩიძე

ტექნიკის მეცნიერებათა კანდიდატი

აკადემიური დოქტორი

თბილისი, 2013

ნინო შენგელია
“მოთხოვნების დამუშავება და ოპტიმიზაცია”

სამაგისტრო ნაშრომის მიზანია მონაცემთა ბაზის სისტემებში მოთხოვნის დამუშავებისა და ოპტიმიზაციის ფუნდამენტალური მეთოდების შესრულების თანმიმდევრული განხილვა და შესწავლა. განსაკუთრებული აქცენტი კეთდება მოთხოვნების გამოთვლის პროცედურების ეფექტურობის გაზრდის სტრატეგიებზე. ნაშრომში განხილული საკითხები აქტუალური და მნიშვნელოვანია როგორც არსებული მონაცემთა ბაზების მართვის სისტემებისათვის ასევე ბაზებთან მომუშავე ფართო აუდიტორიისათვის, მონაცემთა ბაზის სისტემების დამპროექტებლებისა და ადმინისტრატორებისათვის, დანართების შემმუშავებლებისათვის და ამ დანართების მომხმარებლებისთვისაც კი.

Nino Shengelia

“Query Processing and Optimization”

This Master's paper aims to rethink and study sequential performance of query processing and optimization fundamental methods in database systems. Particular emphasis is placed on strategies for increasing the efficiency of calculation procedures of queries. The issues discussed in this paper are relevant and important not only for existing database management systems but also for a wide audience of people who works with databases as database system planners and managers, application developers and those who even works with applications.

სარჩევი

შესავალი.....	4
1. მოთხოვნების ოპტიმიზაციის პრობლემა.....	7
1.1 მოთხოვნა	7
1.2 ოპტიმიზაციის მიზნები	8
1.3 მოთხოვნების ოპტიმიზაციის დადამავალი მიდგომა	12
2. მოთხოვნების წარმოდგენა.....	14
2.1 რელიაციური გამოთვლა.....	14
2.2 რელიაციური ალგებრა	17
2.3 მოთხოვნების გრაფები.....	19
2.4 ტაბლო.....	20
3. მოთხოვნების გარდაქმნა	23
3.1 სტანდარტიზაცია.....	23
3.2 გამარტივება	27
3.3 გაუმჯობესება	30
4. მოთხოვნების გამოთვლა.....	34
4.1 გამოსახულება ერთი ცვლადით.....	34
4.2 გამოსახულებები ორი ცვლადით.....	36
4.3 გამოსახულებები რამდენიმე ცვლადით.....	39
5. წვდომის გეგმები.....	42
5.1 წვდომის გეგმების გენერაცია	43
5.2 წვდომის გეგმების ღირებულების ანალიზი.....	44
5.3 წვდომის გზების არჩევა.....	45
5.4 რამოდენიმე მოთხოვნის მხარდაჭერა	46
დასკვნა	48

შესავალი

მოთხოვნების ოპტიმიზაცია წარმოადგენს ყველაზე მნიშვნელოვან და საინტერესო კვლევის და შემუშავების მიმართულებას მონაცემთა ბაზების სფეროში. ამ მიმართულების შესაძლებლობა განისაზღვრება იმით, რომ მოთხოვნების ოპტიმიზაციის კომპონენტების განვითარებაზე კრიტიკულადაა დამოკიდებული ნებისმიერი SQL ორიენტირებული მონაცემთა ბაზის მართვის სისტემის საერთო წარმადობა (მე შემოვიფარგლები სისტემების ამ კლასით, რადგან პირველ რიგში ისინი თანამედროვე მონაცემთა ბაზების მართვის სისტემების ბაზარზე დომინირებენ და მეორე მხრივ ოპტიმიზაციის მეთოდები ყველაზე მეტად არის განვითარებული სახელდობრ SQL-მოთხოვნებისათვის). ამ მიმართულებას, მე ყველაზე საინტერესოდ მივიჩნევ, რადგან ოპტიმიზაციის ამოცანების გადაწყვეტისას გამოთვლითი მეცნიერებების და მათემატიკის სხვადასხვა სფეროდან ყველაზე განსხვავებული მიდგომებისა და მეთოდების გამოყენება ხდება საჭირო : პროგრამირების ენების კომპილატორში გამოყენებული პროგრამების ოპტიმიზაციის მეთოდები, მათემატიკური ლოგიკა, მათემატიკური სტატისტიკა, ხელოვნური ინტელექტის მეთოდები, გამოსახულების გამოცნობა და ა.შ.

ბოლო ოცდაათი წლის განმავლობაში, ეს ფაქტორები, ამ მიმართულებასთან ასობით მკვლევარს იზიდავდა, რომელთაც ათასობით სტატია აქვთ გამოქვეყნებული, რომელთა უმეტესობა მისაწვდომია და/ან საინტერესოა მხოლოდ პროფესიონალებისათვის. თუმცა, მოთხოვნების ოპტიმიზაციაზე გარკვეული წარმოდგენის ქონა გაცილებით ფართო აუდიტორიისათვის არის სასარგებლო: მონაცემთა ბაზის სისტემების დამპროექტებლებისა და ადმინისტრატორებისათვის, დანართების შემუშავებლებისათვის და ამ დანართების მომხმარებლებისთვისაც კი. სწორედ ამ სახის ცოდნას აწვდის მოცემული ნაშრომი მკითხველს მოთხოვნების ოპტიმიზაციაზე, რადგანაც მოთხოვნების ოპტიმიზაცია ძალიან ფართო თემაა ამ ნაშრომში განხილული იქნება მოთხოვნების დამუშავებისა და ოპტიმიზაციის ფუნდამენტალური მეთოდების პროცესის თანმიმდევრული სურათი. ეს პროცესი არა რომელიმე კონკრეტულ ენაზე არამედ აბსტრაქტულად რელიაციური გამოთვლების ენაზე იქნება ასახული.

მონაცემთა ბაზების სისტემების კვლევის ორი ძირითადი სფერო არსებობს. ერთ-ერთი მათგანია მონაცემთა მოდელების ანალიზი, რომლებზეც შესაძლოა გამოსახულ იქნას რეალური სამყარო და რომელთა მეშვეობითაც შესაძლოა შეიქმნას ინტერფეისები მომხმარებლების განსხვავებული ტიპებისათვის. ასეთი კონცეპტუალური მოდელები მოიცავენ იერარქიულ,

ქსელურ, რელიგიურ მოდელებს და ასევე მთელ რიგ სემანტიკაზე ორიენტირებულ მოდელებს, რომლებიც დიდი რაოდენობით წიგნებში და სტატიებში განიხილებოდა.

ინტერესების მეორე სფერო შეეხება მბმს-ის უსაფრთხო და ეფექტურ რეალიზაციას. კომპიუტიზირებული მონაცემები ორგანიზაციების ცენტრალური რესურსი ხდება. ეს გათვალისწინებული უნდა იყოს ყოველ რეალიზაციაში, რომელიც საწარმოო გამოყენებისათვის არის დანიშნული, პარალელური წვდომის დროს მონაცემების უსაფრთხოების გარანტირებით, აღდგენითა და რეორგანიზაციით. ერთ-ერთი ყველაზე ძირითადი კრიტიკული შენიშვნა მრავალ ადრეულ მბმს-ასთან მდგომარეობდა შეთავაზებულ ან მძლავრ ოპერაციის დამუშავების დროს ეფექტურობის არ არსებობაში, განსაკუთრებით, მოთხოვნების მეშვეობით მონაცემებთან წდომის შემთხვევაში მათი შიგთავსის მიხედვით. მოთხოვნების ოპტიმიზაცია გამიზნულია ამ პრობლემის გადასაწყვეტად დიდი რაოდენობით მეთოდებისა და სტრატეგიების ინტეგრაციის გზით, რომლებიც მოთხოვნების ლოგიკური გარდაქმნილებიდან დაწყებული წვდომის გზების ოპტიმიზაციისა და ფაილური სისტემების დონეზე მონაცემების შენახვას მოიცავს.

ტრადიციულად ყოველ ამ მიდგომაში გამოიყენებოდა ცალკეული ენა. შესაძლოა, სწორედ ეს წარმოადგენს ერთ-ერთ მიზეზს იმისა, რომ ჯერ კიდევ არ არის წარმოდგენილი მოთხოვნების ოპტიმიზაციის მეთოდების ამომწურავი მიმოხილვა. ნაშრომის მიზანია მოთხოვნების ოპტიმიზაციის მეთოდების წარმოდგენა რელიგიური გამოთვლების ზოგად კარკასში. ნაჩვენებია, რომ რელიგიური გამოთვლები ტექნიკურად რელიგიური ალგებრის წარმოდგენის ექვივალენტურია და გაფართოების განხორციელების საშუალებას იძლევა ქსელური მბმს-ისათვის. გარდა ამისა ბევრი პოპულარული მოთხოვნების ენები როგორიცაა SQL და QUEL ადვილად აისახება რელიგიურ გამოთვლებში.

პირველ რიგში ცენტრალიზებულ მბმს-ში მოთხოვნის ოპტიმიზაციაზეა გამახვილებული ყურადღება. ცენტრალიზებული მოთხოვნების ოპტიმიზაცია მნიშვნელოვანია არა მხოლოდ მეინფრეიმებზე არსებულ ბევრ მონაცემთა ბაზებში - ასევე ბოლო დროს მიკროპროცესორულ მონაცემთა ბაზებში- თუმცა ასევე წარმოადგენს დანაწილებულ სისტემებში მოთხოვნების ოპტიმიზაციის ქვე-პრობლემას. თვითონ დანაწილებული მოთხოვნების ოპტიმიზაციას მხოლოდ მცირედ შეეხებით, ხოლო შემდეგ ორ მონათესავე თემას კი გვერდს ავუვლით:

სამომხმარებლო ოპტიმიზაცია: საინფორმაციო სისტემის საერთო ღირებულება დგინდება მბმს-ის ღირებულებითა და ამ სისტემასთან მუშაობაზე მომხმარებლის ძალისხმევის ღირებულებით. ამ ორ სფეროს შორის საზღვარი მდგომარეობს მოთხოვნების ენის გამოყენების

ფუნქციონალურ შესაძლებლობებში და მოსახერხებლობაში და ყველაზე მნიშვნელოვან მხასიათებელს წარმოადგენს სისტემის რეაგირების დრო. თუ დავუშვებთ რომ მოთხოვნების ენა შეიცავს მითითებულ ფუნქციონალურ შესაძლებლობებს, ხოლო რეაგირების დროის მინიმიზაცია წარმოადგენს მოთხოვნების შესრულების სისტემის მიზანს მაშინ მოთხოვნების ოპტიმიზაცია შესაძლოა ინტერპრეტირებულ იქნას როგორც სამომხმარებლო სისტემის ცალკეული ქვე-პრობლემა.

ფაილების სტრუქტურა: მოთხოვნის ოპტიმიზაციის ალგორითმი უნდა აკეთებდეს არჩევანს მოთხოვნის შესრულებისათვის მრავალ წვდომის გზას შორის. ამ წვდომის გზების რეალიზაციის შიდა დეტალები და შესაბამისი შეფასების ფორმების დასკვნები ნაშრომში არ განიხილება

ნაშრომი შედგება ხუთი თავისაგან, პირველ თავში მოთხოვნის ოპტიმიზაციის გლობალური კარკასია წარმოდგენილი მეორე თავში შედარებულია მოთხოვნის წარმოდგენის ოთხი მეთოდი მათი ოპტიმიზაციისათვის ვარგისიანობის თვალსაზრისით. მესამე თავში ერთ-ერთ მეთოდს რელიაციურ გამოთვლას ვიყენებთ ლოგიკაზე დაფუძნებული გარდაქმნის წარმოსადგენად განვითარებადი სემანტიკური ოპტიმიზაციის მეთოდის ჩათვლით.

გარდაქმნის შემდეგ მოთხოვნა უნდა აისახოს ბრძანებების თანმიმდევრობად რომლებიც მოთხონილ მონაცემებს დააბრუნებს. მეოთხე თავში ვაანალიზებთ ასეთი ოპერაციების რეალიზაციას მონაცემთა შენახვისა და წდომის გზების დაბალ დონიან სისტემაში. მეხუთე თავში წარმოვადგენთ ოპტიმიზაციის პროცედურებს ამ ოპერაციების ინტერგრაციისათვის წდომის გლობალურ ოპტიმალურ გეგმაში.

1. მოთხოვნების ოპტიმიზაციის პრობლემა

მოთხოვნების გამოთვლების პროცედურების ზუსტი ოპტიმიზაცია ზოგადად, გამოსათვლელად ძალიან რთულია და კიდე უფრო მეტ სირთულეს, მონაცემთა ბაზაზე ზუსტი სტატისტიკური ინფორმაციის არ არსებობა ქმნის. მოთხოვნის გამოთვლის ალგორითმებში მნიშვნელოვანწილად ევრისტიკაზე დაყრდნობა გვიწევს. თუმცა, ნაშრომში ტერმინი “მოთხოვნების ოპტიმიზაცია” გამოყენებულ იქნება მოთხოვნების გამოთვლის პროცედურების ეფექტურობის გაზრდის სტრატეგიების აღსანიშნავად. ამ თავში ჩამოყალიბებული იქნება მოთხოვნის ოპტიმიზაციის მიზნები და წარმოდგენილი იქნება ზოგადი პროცედურა შემუშავებული გადაწყვეტილების ძებნის პროცესის სტრუქტურიზაციისათვის.

1.1 მოთხოვნა

მოთხოვნა- ენობრივი ექსპრესიაა, რომელიც აღწერს მონაცემებს, რომელიც თავის მხრივ მონაცემთა ბაზიდან გარკვეულ შერჩევას ექვემდებარება. მოთხოვნების ოპტიმიზაციის კონტექსტში ხშირად იგულისხმება, რომ მოთხოვნები გამოიხატება შიგთავსზე (შინაარსზე) დაფუძნებული მანერით (ხშირ შემთხვევაში ორინტირებულს გარკვეულ სიმრავლეზე, რაც ოპტიმიზატორს გამოთვლის შესაძლო პროცედურებს შორის არჩევანის საშუალებას აძლევს)

მოთხოვნები რამოდენიმე გარემოში გამოიყენება. ყველაზე ცხადია დანართი, რომელშიც საბოლოო მომხმარებლის პირდაპირი მოთხოვნები ხვდება, რომელიც საჭიროებს ინფორმაციას მონაცემთა ბაზის სტრუქტურასა და შიგთავსზე. თუ, მომხმარებლის მოთხოვნილებები შეზღუდულია სტანდარტული მოთხოვნების ნაკრებით, მათი ოპტიმიზირება შესაძლოა ხელით მოხდეს შესაბამისი ძებნის პროცედურების პროგრამირების გზით და მომხმარებლის მიერ შეყვანის შეზღუდვით ფორმატირების მენიუს საშუალებით. თუმცა, თუ გვესაჭიროება გაუთვალისწინებელი მოთხოვნების მიცემა საერთო დანიშნულების მოთხოვნების ენის გამოყენებით, მოთხოვნების ატომატური ოპტიმიზაციის სისტემა ხდება საჭირო.

მოთხოვნების მეორე გამოყენება ხდება ტრანზაქციებში, რომლებიც შენახული მონაცემების ცვლილებას ახდენენ მათი მიმდინარე მნიშვნელობებით (მაგალითად, “საწარმოს ყველა მუშაკს ხელფასი 10%-ით გაეზარდოს”) საბოლოოდ, მოთხოვნების მსგავსი გამოსახულებები, შესაძლოა მონაცემთა ბაზების მართვის სისტემის შიგნით იქნას გამოყენებული მაგალითად, წვდომის უფლების შესამოწმებლად, მთლიანობის შეზღუდვის მხარდასაჭერად და პარალელური წვდომის კორექტული სინქრონიზაციისათვის.

1.2 ოპტიმიზაციის მიზნები

ეკონომიური პრინციპი მოითხოვს, რომ ოპტიმიზაციის პროცედურებმა რესურსების მითითებული რაოდენობისათვის ან ეცადონ გამტარუნარიანობის თვისების მაქსიმიზაცია მოახდინონ ან კი რესურსების გამოყენების მინიმალიზაციას ეცადონ მიმდინარე გამტარუნარიანობის თვისებით. მოთხოვნების ოპტიმიზაცია მიმართულია მოცემული მოთხოვნისათვის და მოთხოვნების ტიპების ნაერთისათვის გამოპასუხების დროის მინიმიზაციაზე მიმდინარე სისტემურ გარემოში. ეს საერთო მიზანი უშვებს მთელ რიგ განსხვავებულ ოპერაციულ მიზნობრივ ფუნქციებს. გამოპასუხების დრო წარმოადგენს გონივრულ მიზანს მხოლოდ იმ შემთხვევაში, თუ ვივარაუდებთ, რომ მომხარებლის დრო წარმოადგენს ყველაზე მნიშვნელოვან კრიტიკულ რესურს. წინააღმდეგ შემთხვევაში შეიძლება ვეცადოთ ტექნიკური რესურსების მოხმარების ღირებულების უშუალო მინიმიზაციას. საბედნიეროდ, ორივე მიზანი გარკვეულწილად ურთიერთთავსებადია. კონფლიქტების წარმოქმნისას უმეტესად ხდება შეზღუდვების დაწესება ხელმისაწვდომ ტექნიკურ რესურსებზე (მაგალითად ძირითად მეხსიერებაში ბუფერული მეხსიერების ზომაზე)

რათა ეფექტურობის სამართლიანი შედარება მოვახდინოთ, იმ სისტემებში, რომელთა შედარებასაც ვახდენთ, მოთხოვნების შესრულების ფუნქციონალური შესაძლებლობები მსგავსი უნდა იყოს. კოდის მიერ [Codd 1972] მოფიქრებული "რელიაციური სიმრთელის" მოთხოვნა კვაზისტანდარტი გახდა. ამ ნაშრომში განხილული მეთოდები, წარმოდგენილია როგორც შენატანები მოთხოვნების რეალიზაციაში, რელიაციურად მთელ ენაზე, მინიმალური შესრულების ღირებულებით და გამოპასუხების დროით. საერთო ღირებულება, რომელიც მინიმიზაციას ექვემდებარება, შემდეგი კომპონენტებისაგან შედგება:

კომუნიკაციის ღირებულება: მონაცემების გადაცემის ხარჯები იმ ადგილიდან, სადაც ისინი ინახება იქ სადაც, ხდება გამოთვლების ჩატარება და რეზულტატის წარდგენა. ეს ღირებულება შედგება კომუნიკაციის არხის ღირებულებისაგან, რომელიც ჩვეულებრივ დაკავშირებულია დროსთან, რომლის განმავლობაშიც არხი ღიაა და გადაგზავნასთან დაკავშირებულ შეყოვნებების ღირებულების ხარჯზე. ეს უკანასკნელი კომპონენტი, მნიშვნელოვანია მოთხოვნის ოპტიმიზაციისათვის და ხშირად ეყრდნობა წრფივ ფუნქციას გადასაცემი მონაცემების მოცულობით.

მეორად მეხსიერებასთან წვდომის ღირებულება: მეორად მეხსიერებიდან ძირითადად მონაცემთა გვერდების ჩატვირთვის ღირებულება (ან დრო). ამ ღირებულებაზე გავლენას ახდენს

შერჩეული მონაცემის სიდიდე (უმთავრეს წილად, შუალედური მონაცემების სიდიდე), ფიზიკურ გვერდებზე მონაცემების კლასტერიზაცია, ხელმისაწვდომი ბუფერული მეხსიერების სიდიდე და გამოყენებული ინსტრუმენტების სიჩქარე.

შენახვის ღირებულება: მეორადი მესიერების და ძირითადი მეხსიერების ბუფერების დაკავების ღირებულება. შენახვის ღირებულება მხოლოდ იმ შემთხვევაშია შესაფერისი, როდესაც მეხსიერება ხდება სისტემის ვიწრო ადგილი, და მოთხოვნილი მეხსიერების ზომა შესაძლოა ერთი მოთხოვნიდან მეორემდე იცვლებოდეს.

გამოთვლის ღირებულება: ცენტრალური პროცესორის გამოყენების ღირებულება (ან დრო).

მოთხოვნების ოპტიმიზაციის ალგორითმების სტრუქტურაზე ამ ღირებულებების კომპონენტებს შორის თანაფარდობა მოქმედებს. ტერიტორიალურად დაყოფილ მბმს-ში, სადაც შედარებით შენელებული კომუნიკაციური არხებია, კომუნიკაციური შეფერხებების ღირებულება დომინირებს, ხოლო სხვა ფაქტორები მნიშვნელოვანია მხოლოდ ლოკალური ქვე-ოპტიმიზაციისათვის. ცენტრალიზებულ სისტემებში მეორად მეხსიერებასთან წვდომის დრო დომინირებს, თუმცა რთული მოთხოვნებისათვის შეიძლება ასევე საკმაოდ მაღალი იყოს ცენტრალური პროცესორის ღირებულებაც. ლოკალურად დაყოფილ მბმს-ებში ყველა ფაქტორს თითქმის მსგავსი წონა გააჩნია, რასაც მივყევართ ძალიან რთულ შეფასების ფუნქციებსა და ოპტიმიზაციის პროცედურებთან.

რადგანაც, ამ ნაშრომში ცენტრალიზირებულ მონაცემთა ბაზებს განვიხილავ, კომუნიკაციის ღირებულება არ ითვალისწინება რადგანაც ასეთ სისტემებში კომუნიკაციური მოთხოვნები არ არის დამოკიდებული მოთხოვნის შესრულების სტრატეგიაზე. ერთეულოვანი მოთხოვნების ოპტიმიზაციისათვის შენახვის ღირებულება ასევე არა პირველხარისხოვანია. ეს ხარჯები გაითვალისწინება მხოლოდ რამოდენიმე მოთხოვნის ერთდროული ოპტიმიზაციისას.

რჩება მეორად მეხსიერებასთან წვდომის ღირებულება (რომელიც, ჩვეულებრივ გვერდებთან მიმართვის რაოდენობით განისაზღვრება) და ცენტრალური პროცესორის გამოყენების ღირებულება (რომელიც, ხშირად იმ შედარებების რაოდენობის მიხედვით განისაზღვრება, რომელთა ჩატარებაც მოითხოვება). ამ ღირებულებების შემცირებისათვის შემუშავებული მეთოდების უმეტესეობას საფუძვლად უდევს საერთო იდეების მთელი რიგი: (1) ძალისხმევის დუბლირების თავიდან აცილება (2) სტანდარტიზირებული კომპონენტების გამოყენება (3) წინდახედულობა, რათა თავიდან აცილებულ იქნას ზედმეტი ოპერაცია (4) ელემენტარული ოპერაციების შესრულებისათვის ყველაზე იაფფასიანი საშუალებების გამოყენება (5) მათი

რიგითობის ოპტიმალური სახით დალაგება. შემდეგი მარტივი მაგალითი აჩვენებს თუ რას შეიძლება მოველოდით მოთხოვნების ოპტიმიზაციისაგან.

განვიხილოთ, მონაცემთა ბაზის რელიაციური სქემა, რომელიც აღწერს თანამშრომლებს, რომლებიც კომპიუტერულ ლექციებს სთავაზობენ ორგანიზაციის გეოგრაფიულად განაწილებულ განყოფილებებში:

```
employees (enr, ename, status, city)
papers (enr, title, year)
departments (dname, city, street address)
courses (cnr, cname, abstract)
lectures (cnr, dname, enr, daytime)
```

საგასაღებო ატრიბუტები ხაზგასმულია. საგასაღებო ატრიბუტების მოცემული კომბინაცია დამოკიდებულების ელემენტის უნიკალურ იდენტიფიცირებას ახდენს. ჩავთვალოთ, რომ მომხმარებელს აინტერესებს „ნიუორკში მდებარე იმ განყოფილებების დსახელებები, რომლებიც მონაცემთა ბაზების მართვის კურსს გვთავაზობენ“

არსებობს ამ მოთხოვნის გადაწყვეტის მრავალი შესაძლო სტრატეგია, რომელთაგან სამის შედარება, შემდეგ მოსაზრებებთან, მონაცემთა რეალური მნიშვნელობებისათვის ხდება. აღსანიშნავია, რომ დეტალური მონაცემები რომლებიც ქვემოთ მოყვანილ გამოთვლებში გამოიყენება ჩვეულებრივ მოთხოვნის ოპტიმიზატორისათვის მიუწვდომელია და მხოლოდ უნდა შეფასდეს.

არსებობს 100 განყოფილება, რომელთაგან 5 ნიუორკში მდებარეობს. ფიზიკურ ბლოკში შესაძლებელია მოთავსდეს 5 ჩანაწერი განყოფილებებზე ან dname 50 მნიშვნელობა.

არსებობს 500 კურსი, რომელთაგან 20 მონაცემთა ბაზის მართვას შეეხება. ფიზიკურ ბლოკში შესაძლებელია მოთავსდეს 10 ჩანაწერი.

არსებობს 2000 ლექცია, რომელთაგან 300 შეეხება მონაცემთა ბაზის მართვას, 100 ნიუორკის განყოფილებებში ტარდება ხოლო 20 (3 განყოფილებიდან) ორივე პირობას აკმაყოფილებს. ფიზიკურ ბლოკში შესაძლებელია მოთავსდეს 10 ჩანაწერი

ასევე ვივარაუდოთ, რომ სორტირების დრო შეადგენს $N * \log(2)N$, სადაც N არის ბლოკებში ფაილის ზომა და რომ არსებობს ერთ ბლოკიანი ბუფერი ყოველი დამოკიდებულებისათვის.

საბოლოოდ ყველა ურთიერთობა ფიზიკურად დალაგებულია გასაღების მნიშვნელობის ზრდადობის მიხედვით.

პირველი, აქ მოყვანილი სტრატეგია, მიჰყვება რელიაციური გამოთვლის გამოსახულების ტრანსლაციის პირდაპირ მიდგომას ალგებრის ოპერაციებში. სტრატეგიის ყოველ ნაბიჯთან ერთად მოყვანილია მეორად მეხსიერებასთან მიმართვის რაოდენობა, რომელიც საჭიროა წასაკითხად (r) და ჩასაწერად (w).

სტრატეგია 1

1. "courses", "lectures" და "departments" დამოკიდებულებების დეკარტული გამოსახულების ფორმირება (r : 200000)
2. $dname$ სვეტის დატოვება "departments" იმ ჩანაწერებიდან, რომელთათვისაც cnr სვეტების მნიშვნელობები "courses" და "lectures"-ში ერთმანეთს ემთხვევა და $dname$ -ის სვეტების მნიშვნელობები "lectures" და "departments"-იდან ერთმანეთს ემთხვევა და $cname = 'database management'$ and $city = 'New York'$. (w : 1)

ჯამში: დაახლოებით 200000 მიმართვა.

ეს სტრატეგია ძალიან ძვირი იმ ფაქტის გამოა, რომ ის შუალედურ რეზულტატს აგენერირებს, რომლის მხოლოდ ძალიან მცირე ნაწილია მართლაც საჭირო შემდეგი დამუშავებისათვის. ეფექტურობა შეიძლება არსებითად გაიზარდოს მხოლოდ იმ განსხვავებული დამოკიდებულებების ელემენტების კომბინაციების განხილვით, რომლებსაც საერთო ატრიბუტების ერთნაირი მნიშვნელობები გააჩნიათ. სორტირების არსებული თანმიმდევრობის გამოყენების გზით ასეთი კომბინაციები შეიძლება ფორმირებულ იქნას მონაწილე დამოკიდებულებების „შერწყმით“. ამ ოპერაციას „გაერთიანება“ ეწოდება.

სტრატეგია 2

1. "courses" და "lectures" დამოკიდებულებების შერწყმის შესრულება (r : 50 + 200; w : 400)
2. $dnames$ -ის მიხედვით რეზულტატის სორტირება ($r + w$: $400 * \log(2)400$)
3. შედეგის შერწყმის შესრულება "departments"-ს დამოკიდებულებასთან (r : 400 + 20; w : 400 + 400)
4. ავირჩიოთ კომბინაცია $cname = 'database management'$ and $city = 'New York'$ (r : 800)
5. მხოლოდ $dname$ სვეტის დატოვება. (w :1)

ჯამში: დაახლოებით 6000 მიმართვა.

მოთხოვნაზე პასუხის ღირებულება შეიძლება კიდევ უფრო შემცირდეს, რაც შეიძლება ადრე მნიშვნელობების საფუძველზე შეზღუდვების შესრულების გზით და ამის ხარჯზე სორტირებაზე და შუალედური რეზულტატების შერწყმაზე ხარჯების შემცირებით.

სტრატეგია 3

1. "courses" და "lectures" დამოკიდებულებების შერწყმის შესრულება ($r: 50 + 200$)
2. `cname = 'database management'` კომბინაციიდან მხოლოდ `dnames` დავტოვოთ ($w: 2$)
3. დავასორტიროთ გენერირებული `dname` სია ($r + w: 2$)
4. რეზულტატის შერწყმა შევასრულოთ "departments" დამოკიდებულებასთან ($r: 2 + 20$)
5. მხოლოდ ის `dnames` დავტოვოთ რომელთათვისაც `city = 'New York'` ($w: 1$)

ჯამში: დაახლოებით 277 მიმართვა.

ამგვარად, მივაღწიეთ ღირებულების შემცირება დახლოებით 700-ჯერ. დიდი მონაცემთა ბაზებისათვის და უფრო რთული მოთხოვნებისათვის უფრო მეტად სრულყოფილი მეთოდების გამოყენებით შესაძლებელია ღირებულების კიდევ უფრო მეტად შემცირება.

1.3 მოთხოვნების ოპტიმიზაციის დადმავალი მიდგომა

ლიტერატურაში არსებული მოთხოვნების ოპტიმიზაციის კვლევები შესაძლებელია ორ კლასად დაიყოს, რომელთა დახასიათება შეიძლება როგორც აღმავალისა და დადმავალის. მკვლევარები მოთხოვნების ოპტიმიზაციის ზოგად პრობლემას ძალიან რთულად მიიჩნევენ. თეორიული სამუშაოები აღმავალი მიდგომით დაიწყო, განსაკუთრებული შემთხვევების შესწავლა როგორებიცაა მოთხოვნების გარკვეული ქვეკლასებისათვის მნიშვნელოვანი ოპერაციების და გამოთვლითი სტრატეგიების ოპტიმალური რეალიზაცია. მომავალში მკვლევარები ცდილობდნენ ამ საწყისი მონაცემებიდან უფრო დიდი სამშენებლო ბლოკების შეგროვებას.

მუშა სისტემების საჭიროებამ გამოიწვია მოთხოვნების გამოთვლის სრულმასშტაბიანი პროცედურების შემუშავება, რამაც გავლენა იქონია გადაწყვეტილებების ერთიანობაზე და აიძულა მოთხოვნების ოპტიმიზაცია წარმოებულყო ერთგავროვანი და ევრისტიკული მანერით. რადგანაც, ხშირად ეს სისტემის კონკურენტუნარიანობის ეფექტურობის მიღწევის საშუალებას არ იძლეოდა, თანამედროვე ტენდენციად წარმოგვიდგება დადმავალი მიდგომა, რომელიც საერთო

პროცედურებში კერძო შემთხვევებში ოპტიმიზაციის შესაძლებლობების დიდი ცოდნის ჩართვის საშუალებას უზრუნველყოფს. ამავე დრო, თვითონ საერთო ალგორითმები უმჯობესდებიან სტრატეგიებს შორის არჩევანის გაკეთების ღირებულების მინიმიზაციის კომბინატორული პროცედურებით.

ნაშრომში დაღმავალ მიდგომას ვიყენებ, ვიყენებ რა გამოთვლების საერთო პროცედურას რომელიც მოთხოვნების ოპტიმიზაციის კვლევისას შემუშავებული კონკრეტული მეთოდების კარკასს წარმოადგენს:

ნაბიჯი 1. მოთხოვნის ისეთი შიდა წარმოდგენის მოძებნა, რომელშიც მომხმარებლების მოთხოვნები შესაძლებელია მარტივად აისახოს და რომელიც სისტემას მოთხოვნის შესრულების ოპტიმიზაციისათვის აუცილებელ თავისუფლებას დაუტოვებს.

ნაბიჯი 2. მოთხოვნის წარმოდგენისათვის ლოგიკური გარდაქმნების გამოყენება რომლებიც (1) მოთხოვნას სტანდარტიზირებულს გახდიან (2) გაამარტივებენ მოთხოვნას რათა ძალისხმევის დუბლირება თავიდან ავიცილოთ (3) გააუმჯობესებენ მოთხოვნას მისი შესრულების გასამარტივებლად და კერძო შემთხვევების პროცედურების გამოყენების შესაძლებლობის შესაქმნელად.

ნაბიჯი 3. გარდაქმნილი მოთხოვნის გამოსახვა ელემენტარული ოპერაციების შესაძლო რიგითობად, რომელთათვისაც კარგი რეალიზაცია და შესაბამისი ღირებულების შეფასებებია ცნობილი. ამ ნაბიჯის რეზულტატად წარმოიქმნება შესაძლო „წვდომის გეგმების“ ნაკრები.

ნაბიჯი 4. თითოეული წვდომის გეგმის საერთო ღირებულების გამოთვლა, ყველაზე იაფი გეგმის შერჩევა და შესრულება.

ამ პროცედურის პირველი ორი ნაბიჯი გარკვეულწილად დამოუკიდებელია და ამიტომაც, ხშირად შესაძლოა შესრულებულ იქნას კომპილაციის დროს. მესამე-მეოთხე ნაბიჯების ხარისხი, გენერირებული წვდომის გეგმების სიმრავლე და შერჩევის ალგორითმის ოპტიმალურობა მნიშვნელოვნად არის დამოკიდებული მონაცემთა ბაზაში არსებული მნიშვნელობების ცოდნაზე.

მონაცემებზე დამოკიდებულების შედეგები ორმხრივია. პირველ რიგში, თუ მონაცემთა ბაზა ცვალებადია, მაშინ მესამე და მეოთხე ნაბიჯები შესაძლოა შესრულებულ იქნას მხოლოდ შესრულების დროს. ეს ნიშნავს იმას, რომ შესაძლო მოგება ეფექტურობაში უნდა შეესაბამებოდეს თვითონ ოპტიმიზაციის ღირებულებას. მეორე მხრივ, მონაცემთა მეტაბაზაში (მაგალითად, მონაცემების გაფართოებად სარჩევში) უნდა იყოს მხარდაჭერილი საერთო ინფორმაცია

მონაცემთა ბაზის სტრუქტურაზე, ისევე, როგორც სტატისტიკური ინფორმაცია მონაცემთა ბაზის შიგთავსზე. როგორც ბევრ მსგავს ოპერაციულ კვლევით პრობლემებზე (მაგალითად, მარაგების მართვა) ამ დამატებითი ღირებულების მიღებისა და მხარდაჭერის ინფორმაცია უნდა დარღებოდეს მის ხარისხს.

2. მოთხოვნების წარმოდგენა

მოთხოვნები შესაძლოა წარმოდგენილ იქნას მთელი რიგი სახით. მოთხოვნების ოპტიმიზაციის კონტექსტში მოთხოვნების მართებულმა წარმოდგენამ უნდა დააკმაყოფილოს შემდეგი მოთხოვნები: ის უნდა იყოს საკმაოდ მძლავრი მოთხოვნების დიდი კლასის გამოსახატავად და უნდა უზრუნველყოს სწორად აწყობილი ბაზისი მოთხოვნების გარდასაქმნელად. ამ თავში აღწერილი იქნება მოთხოვნის წარმოდგენის ოთხი განსხვავებული ფორმა, რომელთაგან თითოეული გამოიყენება მოთხოვნების ოპტიმიზაციამდე მისასვლელად.

2.1 რელიაციური გამოთვლა

(კორტეჟული) რელიაციური გამოთვლა, როგორც კოდის მიერ იქნა შემოტანილი [Codd 1971, 1972] -წარმოადგენს ნოტაციას მოთხოვნის რეზულტატის განსასაზღვრად მისი თვისებების აღწერის გზით. მოთხოვნის წარმოდგენა რელიაციური გამოთვლის სახით შედგება ორი ნაწილისაგან: სამიზნე სია (target list) და შერჩევის გამოსახულება (selection expression).

შერჩევის გამოსახულება ახდენს მოთხოვნის შესრულების რეზულტატად შექმნილ დამოკიდებულების შიგთავსის სპეციფიცირებას, პირველი რიგის პრედიკატების მეშვეობით. (ანუ განზოგადოებულ ბულის ტიპის გამოსახულებებით, რომლებიც შესაძლოა არსებობის და უნიკრსალობის კვანტორებს შეიცავენ). სამიზნე სიაში განისაზღვრება თავისუფალი ცვლადები, რომლებიც პრედიკატში გვხვდებიან და შედეგად მიღებული დამოკიდებულების სტრუქტურის სპეციფიცირება სრულდება. 2.1 მაგალითში დემონსტრირებულია რელიაციური გამოთვლის წარმოდგენა Pascal/R მონაცემთა ბაზის პროგრამირების ენის სინტაქსის გამოყენებით

მაგალითი 2.1 *იმ პროფესორების სახელები რომელთაც რაიმე სახის სტატია გამოაქვეყნეს 1981 წელს*

```
[<e.ename> OF  
EACH e IN employees:  
    e.status = professor
```

AND

SOME p IN papers

(e.enr = p.enr AND p.year = 1981)]

სამიზნე სიაში ანუ ქვეგამოსახულებაში რომელიც ორწერტილამდეა, (თავისუფალი) ცვლადი e-ს განსაზღვრის არე შემოისაზღვრება "employees" დამოკიდებულების ელემენტებით. ამიტომაც, დამოკიდებულება "employees"-ს ეწოდება e-ცვლადის განსაზღვრის არის დამოკიდებულება (range relation). მიზნის ატრიბუტის სპეციფიკაცია "<e.ename>" აჩვენებს, რომ მოთხოვნის რეზულტატად დარჩება მხოლოდ თანამშრომლების სახელები.

შეჩვევის გამოსახულება - პრედიკატი, რომელიც ორწერტილს მოჰყება - განსაზღვრავს შეზღუდვებს თავისუფალ ცვლადზე. პირველი შეზღუდვა წარმოადგენს შემზღუდველ ანუ ერთადგილიან ტერმს (monadic term), რომელიც თავისუფალ ცვლადს ზღუდავს იმ "employees" ჩანაწერებით რომელთა status-სვეტის მნიშვნელობაა "professor". ეს შეზღუდვა დაკავშირებულია AND კავშირით შემაერთებელ ან ორადგილიან ტერმთან (dyadic term), რომელიც აკავშირებს "employees" -ს "papers"-თან და კიდევ ერთ ერთადგილიან ტერმთან, რომელიც კიდევ უფრო ზღუდავს რეზულტატს იმ თანამშრომლებით, რომელთაც რაიმე სტატია მაინც აქვთ გამოქვეყნებული 1981 წელს. ჩვეულებრივ, ტერმებში დასაშვებ შედარების ოპერაციებს წარმოადგენს =, <, >, d და e.

განსხვავებით პრედიკატების ერთკლასიანი გამოთვლებისაგან, რელიაციურ გამოთვლებში დაიშვება ცვლადები, რომლებიც დაკავშირებულია განსხვავებულ კლასებთან (განსაზღვრის არის დამოკიდებულებებთან) მაგალითად, ცვლადი e დაკავშირებულია "employees"-თან ხოლო ცვლადი p "papers"-თან. რელიაციური გამოთვლების მრავალკლასობრიობის შედეგი მოთხოვნის გარდაქმნასთან მიმართებაში განიხილება 3.1 თავში.

გარდა ლოგიკური AND ოპერაციისა, პრედიკატებში შეიძლება გამოყენებულ იქნას ასევე OR და NOT ოპერაციები. რელიაციური გამოთვლების პრედიკატები სრულიად განისაზღვრება შემდეგი რეკურსიული წესებით:

1. ატომარული პრედიკატები

- i. (ერთადგილიანი ან ორადგილიანი) ტერმი წარმოადგენს ატომარულ პრედიკატს.
- ii. TRUE წარმოადგენს ატომარულ პრედიკატს.
- iii. FALSE წარმოადგენს ატომარულ პრედიკატს.

2. ატომარული პრედიკატი წარმოადგენს პრედიკატს. თუ A - პრედიკატია, r - ელემენტის ცვლადი და rel - დამოკიდებულება. მაშინ

- i. $SOME\ r\ IN\ rel(A)$,
- ii. $ALL\ r\ IN\ rel(A)$

ასევე პრედიკატებია.

3. თუ A ან B პრედიკატებია. მაშინ

- i. $NOT\ (A)$ (უარყოფა),
- ii. $A\ AND\ B$ (კონიუნქცია),
- iii. $A\ OR\ B$ (დიზიუნქცია)

წარმოადგენენ პრედიკატებს.

4. არც ერთი სხვა ფორმულა პრედიკატს არ წარმოადგენს.

რელიაციური გამოთვლა შემოტანილ იქნა [Codd 1972] როგორც რელიაციური სიმბლავრის საზომი. წარმოდგენის ფორმას ეწოდება რელიაციურად სრული თუ მასში დაიშვება რელიაციური გამოთვლის ამოსახულების მიერ განსაზღვრული ნებისმიერი მოთხოვნის რეზულტატის განსაზღვრა. ნათელია, რომ რელიაციური სიმრთელე უნდა განიხილებოდეს, როგორც მინიმალური მოთხოვნა გამოსახვით სიმბლავრესთან მიმართებაში. ხშირად, უბრალო კონცეპტუალური მოთხოვნის მაგალითად რომელიც რელიაციური სიმრთელის საზღვრებს სცდება მოყვანილია მოთხოვნა „იმ თანამშრომლების სახელების მოძებნა, რომლებიც მენეჯერი სმიტის წინაშე ნებისმიერ დონეზე აგებენ პასუხს“, ეს მაგალითი ითვალისწინებს, რომ ერთ დამოკიდებულებაში მოდელირდება თანამშრომელთა იერარქია ($name$ და $manager$ ატრიბუტების საშუალებით) გარდა ამისა, მოთხოვნები დღევანდელ დანართებში ხშირად აგრეგაციებს შეიცავენ, რომელთა გამოსახვაც შეუძლებელია წმინდა რელიაციურ გამოთვლებში. თუმცა რელიაციური გამოთვლები საკმაოდ ადვილად ფართოვდება აგრეგატული ფუნქციებით [Klug 1982b; Maier and Warren 1981].

2.2 რელიაციური ალგებრა

რელიაციური ალგებრა, როგორც განსაზღვრულია [Codd 1972] კოდის მიერ წარმოდგენს დამოკიდებულებებზე ოპერაციების კრებულს. ეს ოპერაციები ორ კლასად იყოფა: ტრადიციული ოპერაციები სიმრავლეებზე, როგორიცაა დეკარტული ნამრავლი, გაერთიანება, თანაკვეთა და სხვაობა და რელიაციური ალგებრის სპეციალური ოპერაციები, როგორებიცაა შეზღუდვა, პროექცირება, გაერთიანება და გაყოფა. ქვემოთ მოყვანილი სპეციალური ოპერაციები განისაზღვრება მათი დაკავშირების გზით რელიაციური გამოთვლის ექვივალენტურ გამონათქვამებში.

შეზღუდვის ოპერაცია, რომელიც გამოყენებულია "rel" დამოკიდებულებასთან, ქმნის ჰორიზონტალურ ქვესიმრავლეს არა კვანტორულ პრედიკატთან შესაბამისად, რომელიც მხოლოდ ერთადგილიან ტერმებს ან "დამოკიდებულებას შორის" ორადგილიან ტერმებს შეიცავს (დამოკიდებულების ერთი ელემენტის ატრიბუტებს შორის შედარება):

```
Rest(rel, pred) = [EACH r IN rel: pred].
```

პროექციის ოპერაცია ემსახურება "rel" დამოკიდებულების ვერტიკალური ქვესიმრავლის კონსტრუირებას A მითითებული ატრიბუტების ნაკრების შერჩევით და ამ ატრიბუტებიდან კორტეჟი-დუბლიკატების წაშლით:

```
Proj(rel, A) = [(r.A) OF EACH r IN rel: true].
```

გაერთიანების ოპერაცია საშუალებას იძლევა გავაერთიანოთ ორი დამოკიდებულება "rel₁" და "rel₂" ერთ დამოკიდებულებად, რომლის ატრიბუტები წარმოდგენს "rel₁" და "rel₂"-ის ატრიბუტების გაერთიანებას:

```
Join(rel1, A op B, rel2) =  
[EACH r1 IN rel1, EACH r2 IN rel2: r1.A op r2.B].
```

გაერთიანებების ოპერაციებში დაშვებული "op" შედარება იგივეა რაც რელიაციური გამოთვლის ორადგილიან ტერმებში. თუ "op" ტოლობაზე შედარების ოპერაციას წარმოდგენს, მაშინ "ზუნებრივი" გაერთიანების შედეგად იწევა A ან B.

გაყოფის ოპერაცია წარმოადგენს უნივერსალობის კვანტორის ალგებრულ ასლს. ის შემდეგნაირად განისაზღვრება:

```

Divi( $rel_1$ , A by B,  $rel_2$ ) =
    [ ( $r.compl(A)$ ) OF EACH  $r_1$  IN  $rel_1$ :
      ALL  $rel_2$  IN  $rel_2$  SOME  $r_3$  IN  $rel_1$ 
      ( $r_1.compl(A) = rel_2.compl(A)$  AND  $rel_2.B = r_3.A$ ) ],

```

სადაც $compl(A)$ - ეს არის " rel_1 " ატრიბუტების სიმრავლეში A-ს დამატებაა. როგორც განსაზღვრება გვიჩვენებს, გაყოფა საკმაოდ რთული ოპერაციაა, რასაც მოთხოვნის აღქმის გართულება შეუძლია.

2.2 მაგალითში გამოსახულია 2.1 მაგალითის მოთხოვნა რელიაციური ალგებრის ტერმინებში.

მაგალითი 2.2 *იმ პროფესორების სახელები რომელთაც რაიმე სახის სტატია გამოაქვეყნეს 1981 წ.*

```

Proj (Rest (Join (employees,
                  enr = enr,
                  Rest (papers, year = 1981)),
              status = professor),
      ename)

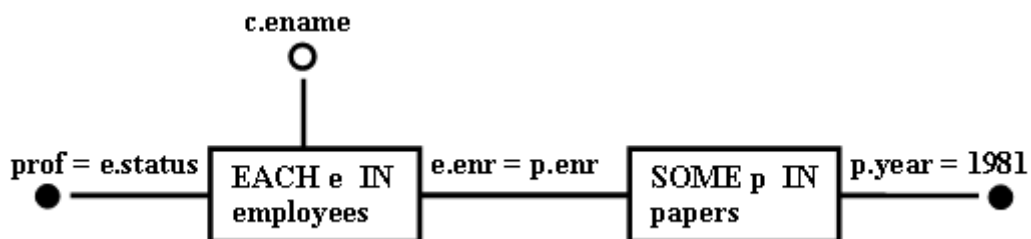
```

რელიაციური გამოთვლის გამოსახულების საპირისპიროდ, რომელიც აღწერს დამოკიდებულებას გამომდინარე მოთხოვნიდან, მისი თვისებების ტერმინებით, რელიაციური ალგებრის გამოსახულება განსაზღვრავს შედეგად მიღებული დამოკიდებულების კონსტრუირების ალგორითმს. გამოთვლის გამოსახულება, როგორც ჩანს მოთხოვნის ოპტიმიზაციისათვის საუკეთესო ამოსავალ წერტილს წარმოადგენს, რადგანაც ის ოპტიმიზატორს მხოლოდ მოთხოვნის საბაზისო თვისებებით უზრუნველყოფს. ოპტიმიზაციის შესაძლებლობები შეიძლება დამალულ იქნას ალგებრის ოპერაციების კონკრეტულ თანმიმდევრობაში. თუმცა, რელიაციურ სიმრთელესთან მიმართებაში რელიაციური ალგებრა ისევე სრულია როგორც რელიაციური გამოთვლა. [Codd 1972]-ში ნაჩვენებია, რომ რელიაციური გამოთვლის ნებისმიერი გამოსახულება შეიძლება გადაითარგმნოს ალგებრის ექვივალენტურ გამოსახულებაში. ანალოგიური რეზულტატი ალგებრის გამოსახულებებისათვის და გაფართოებული აგრეგირებული ფუნქციებით გამოთვლისათვის დამტკიცებულია კლუგის მიერ [Klug 1982a].

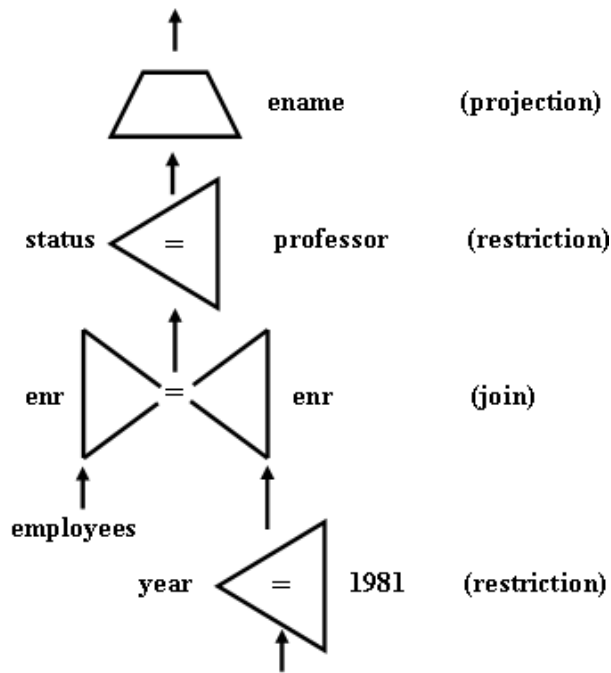
2.3 მოთხოვნების გრაფები

გრაფები გამოიყენება სტრუქტურირებული ობიექტების ვიზუალური წარმოდგენისათვის მთელ რიგ სფეროებში. ორ კარგად ცნობილ მაგალითს წარმოადგენს კომპილატორების კონსტრუირების დროს სინსტაქსური ხეების გამოყენება და AND/OR გრაფების გამოყენება ხელოვნური ინტელექტის დანართებში. მოთხოვნის ოპტიმიზაციისას გრაფები გამოიყენება მოთხოვნებისა და მოთხოვნების გამოთვლის სტრატეგიების წარმოსადგენად. შეიძლება გამოვყოთ გრაფების ორი კლასი: ობიექტების გრაფები და ოპერაციების გრაფები.

ობიექტების გრაფებში კვანძებს წარმოადგენს ობიექტები, როგორებიცაა ცვლადები (დამოკიდებულებების) და კონსტანტები. რკალები აღწერენ პრედიკატებს, რომელთაც ეს ობიექტები უნდა აკმაყოფილებდნენ. ობიექტების გრაფები შეიცავენ მოთხოვნის რეზულტატის თვისებებს და ამიტომაც მჭიდროდ არიან დაკავშირებული რელიაციურ გამოთვლებთან. ოპერაციის გრაფები აღწერენ ოპერაციების მიერ მართვად მონაცემთა ნაკადებს, ოპერაციების როგორც კვანძების წარმოდგენის გზით, რომლებიც რკალებით არიან დაკავშირებული და რომლებიც თავის მხრივ მონაცემთა მოძრაობის მიმართულებას მიუთითებენ. Yao [1979]-ში ოპერაციის გრაფები გამოიყენებოდა ალგებრის გამოსახულების წარმოსადგენად. ნახაზ 1 და ნახაზ 2-ზე მოყვანილია შესაბამისად ობიექტების გრაფებისა და ოპერაციების გრაფების მაგალითები.



ნახაზი 1. ობიექტის გრაფი, რომელიც მაგალითში აღწერილ მოთხოვნას წარმოადგენს



ნახაზი 2. ოპერაციის გრაფი, რომელიც მაგალითში აღწერილ მოთხოვნას წარმოადგენს

მოთხოვნის გრაფებს ბევრი ძალიან კარგი თვისება გააჩნია. მოთხოვნის ვიზუალური წარმოდგენა მისი სტრუქტურული მახასიათებლების უფრო მარტივად აღქმას უწყობს ხელს. გარდა ამისა, გრაფების თეორიაში გრაფების ავტომატური ანალიზისათვის სასარგებლო ბევრი შედეგი არსებობს, მაგალითად ციკლების აღმოჩენა და ხისებრობის თვისება. საბოლოოდ მოთხოვნის გრაფების მნიშვნელოვანი უპირატესობაა ის, რომ მათი გაფართოება დამატებითი ინფორმაციით საკმაოდ ადვილად შეიძლება. მაგალითად, გრაფების გაფართოება მონაცემების ფიზიკური ორგანიზაციის დეტალებით შემოთავაზებულია [Rosenthal and Reiner 1982].

2.4 ტაბლო

ტაბლო, როგორც ეს განისაზღვრა აჰო-ს და სხვების მიერ [Aho et al. 1979a, 1979b, 1979c] წარმოადგენს ისეთი რელიაციური გამოთვლის ქვესიმრავლის ტაბლოურ (ცხრილურ) ნოტაციას, რომელიც ხასიათდება მხოლოდ AND-ით დაკავშირებული ტერმებით და უნივერსალობის კვანტორის არ არსებობით. ამგვარად, ტაბლო-მოთხოვნები წარმოადგენს კონიუქტიურ მოთხოვნების კერძო სახეს [Chandra and Merlin 1977; Rosenkrantz and Hunt 1980].

ტაბლოები წარმოადგენენ სპეციალიზირებულ მატრიცებს, რომელთა სვეტებიც შეესაბამებიან მონაცემთა ბაზის შესაბამისი სქემის ატრიბუტებს. მატრიცის პირველი შემაჯამებელი (summary)

სტრიქონი, იგივე მიზნებს ემსახურება, რომელსაც რელიგიური გამოთვლის გამოსახულების სამიზნე სია. დარჩენილი სტრიქონები კი აღწერენ პრედიკატებს. სიმბოლოები, რომლებიც ტაბლოში ჩნდებიან წარმოადგენენ გამოყოფილ (distinguished) ცვლადებს (შესაბამებთან თავისუფალ ცვლადებს), არა გამოყოფილ (nondistinguished) ცვლადებს (არსებობის კვანტორის ნიშნით მყოფი ცვლადების შესაბამისად), კონსტანტებს, ცარიელი ადგილებს და ტეგებს (რომლებიც, განსაზღვრის სფეროების კავშირებს მიუთითებენ).

ნახაზ 3-ზე ილუსტრირებულია ტაბლოს კონსტრუირება, რომელიც 2.1 მაგალითში მოყვანილ მოთხოვნას წარმოადგენს. პროცესი იწყება ერთ დამოკიდებულებიანი ტაბლოთი და გრძელდება ამ ტაბლოების კომბინირების გზით ახალ ტაბლეობად უფრო მეტად დიდი გამონათქვამებისათვის. გამოყოფილი ცვლადები აღინიშნება a-სიმბოლოთი, ხოლო არა გამოყოფილი აღინიშნება b-თი.

		status	ename	enr	year	
T(employees)	=	a1	a2	a3		employees
		a1	a2	a3		
T(papers)	=			a3	a4	papers
				a3	a4	
T(Rest (papers, year = 1981))	=			a3	1981	papers
				a3	1981	
T(Join (employees, enr = enr, Rest(papers, year = 1981))	=	a1	a2	a3	1981	employees papers
		a1	a2	a3	1981	
T((Rest (Join (employees, enr = enr, Rest(papers, year = 1981) , status = professor))	=	professor	a2	a3	1981	employees papers
		professor	a2	a3	1981	
T(Proj (Rest (Join (employees, enr = enr, Rest(papers, year = 1981) , status = professor) , ename))	=		a2			employees papers
		professor	a2	b3	1981	

ნახაზი 3. T ტაბლოს ეტაპობრივი კონსტრუირება, რომელიც წარმოადგენს 2.1 მაგალითში მოყვანილ მოთხოვნას

გამოსახულება, რომელიც დიზუნქციას (სიმრავლეების გაერთიანებას) შეიცავს და უარყოფას (სიმრავლეების გამოკლებას) შესაძლოა წარმოადგენილ იქნას ტაბლოების ნაკრებით. [Johnson and Klug 1983]-ში ტაბლეობის ნაკრები გამოიყენება საერთო კონიუქტიური მოთხოვნების წარმოსადგენად. მოთხოვნის ოპტიმიზაციის სფეროში ტაბლოს კონკრეტული მნიშვნელობას განვიხილეთ 3.2 თავში.

3. მოთხოვნების გარდაქმნა

როგორც უკვე აღვნიშნეთ, მოთხოვნები შეიძლება გამოისახოს მთელი რიგი განსხვავებული ფორმებით. გარდა ამისა, მითითებულ ენის შიგნითაც კი, ყოველი მოთხოვნისათვის შესაძლოა არსებობდეს რიგი სემანტიკურად ექვივალენტური გამოსახულებებისა. ამ თავში აღიწერება მოთხოვნების გამოსახულების გარდაქმნა ექვივალენტურ გამოსახულებად სწორედ განსაზღვრული წესების მიხედვით. მოთხოვნის გარდაქმნის მიზნები სამგვარია: (1) მოთხოვნის ოპტიმიზაციისათვის სტანდარტიზირებული საწყისი წერტილის კონსტრუირება (სტანდარტიზაცია), (2) სიჭარბის აღმოფხვრა (გამარტივება) და (3) შესრულების ეფექტურობის მიზნით გაუმჯობესებული გამოსახულებების კონსტრუირება (გაუმჯობესება-amelioration)

3.1 სტანდარტიზაცია

მოთხოვნების ოპტიმიზაციის რამოდენიმე მიდგომაში სტანდარტიზირებული საწყისი წერტილი განისაზღვრება მოთხოვნის წარმოდგენისათვის გამოყენებული ფორმის ნორმალიზირებული ვარიანტით [Jarke and Schmidt 1981; Kim 1982; Palermo 1972; Wong and Youssefi 1976]. აქ მოყვანილი იქნება რელიაციური გამოთვლისათვის ორი ნორმალური ფორმა და ასევე წესები რომელთაც უნდა ექვედებარებოდეს ნორმალიზაციის პროცედურა.

იძახიან, რომ რელიაციური გამოთვლის გამოყენებით მოთხოვნის წარმოდგენა წინსაწარ ნორმალურ ფორმაშია (prenex normal form), თუ მისი შერჩევის გამონათქვამი შემდეგი სახისაა

$$\text{SOME/ALL } rel_1 \text{ IN } rel_1 \dots$$
$$\text{SOME/ALL } r_n \text{ IN } rel_n(M),$$

სადაც M - პრედიკატია რომელიც კვანტორებს არ შეიცავს. M -ს მატრიცა (matrix) ეწოდება და ასევე შეიძლება სტანდარტიზირებული იყოს. იძახიან, რომ მატრიცა რომელიც (A_{ij} ტერმების) დიზუნქციისა და კონიუნქციისაგან შედგება, როგორიცაა

$$(A_{11} \text{ AND } \dots \text{ AND } A_{1n}) \text{ OR } \dots$$
$$\text{OR } (A_{m1} \text{ AND } \dots \text{ AND } A_{mn}),$$

დიზუნქციურ ნორმალურ ფორმაში იმყოფება, ხოლო მატრიცა რომელიც დიზუნქციების კონიუნქციისაგან შედგება როგორიცაა

$$A_{11} \text{ OR } . . . \text{ OR } A_{1n}) \text{ AND } . . .$$

$$\text{AND } (A_{m1} \text{ OR } . . . \text{ OR } A_{mn}) ,$$

კონიუნქტიურად ნორმალურ ფორმაში იმყოფება.

წინასწარ ნორმალურ ფორმას მატრიცის ნორმალური ფორმებით მივყავართ რელიაციური გამოთვლის გამონათქვამის ორ ნორმალურ ფორმამდე: დიზუნქტიურ წინასწარ ნორმალურ ფორმამდე (disjunctive prenex normal form, DPNF) და კონიუნქტიურ წინასწარ ნორმალურ ფორმამდე (conjunctive prenex normal form, CPNF). DPNF-ის გამოყენება მოტივირებულია მოთხოვნის დამოუკიდებელი კომპონენტების დაყოფილი ოპტიმიზაციითა და შესრულებით [Bernstein et al. 1981]. CPNF სასარგებლო გამოდგა მოთხოვნების დეკომპოზიციისათვის [Wong and Youssefi 1976] და მონაცემებზე დამოკიდებული მოთხოვნების გაუმჯობესებისათვის (მაგალითად, პირველ რიგში ყველაზე შემზღუდველი დიზუნქტის შემოწმება)

CPNF-ში მოთხოვნები შესაძლოა კიდევ დაექვემდებარონ გარდამნას ისეთ ფორმაში რომელიც კვანტორებს არ შეიცავს ანუ ეგრეთ წოდებულ კლაუზალურ ფორმაში (*clausal form*) [Nilsson 1982] რომელიც პოპულარულია ხელოვნური ინტელექტის თეორემების დანართების დამტკიცებებში. მონაცემთა ბაზის ენები, რომლებიც ლოგიკაზეა დაფუძნებული როგორებიცაა Prolog - ეფუძნებიან კლაუზალურ ფორმას. რადგანაც, კლაუზალური ფორმა იშვიათად გამოიყენება მოთხოვნების ოპტიმიზაციის დროს დეტალები განხილული აღარ იქნება.

რელიაციური გამოთვლის გამოსახულების გარდაქმნა წინასწარ ნორმალურ ფორმაში მდგომარეობს ტერმებში კვანტორების გადაადგილებაში (მარჯვნიდან მარცხნივ). კვანტორების გადაადგილება გარდაქმნის წესებით იმართება, რომელიც ცხრილ 1-შია წარმოდგენილი. მატრიცის ნორმალიზაცია დემორგანის, დისტრიბუციურობისა და ორმაგი უარყოფის წესების გამოყენებით მიიღწევა (ცხრილი 2).

- Q1: $A \text{ AND SOME } r \text{ IN rel } (B(r))$
 $\Leftrightarrow$
 $\text{SOME } r \text{ IN rel } (A \text{ AND } B(r))$
- Q2: $A \text{ OR SOME } r \text{ IN rel } (B(r))$
 $\Leftrightarrow$
 a) $\text{SOME } r \text{ IN rel } (A \text{ OR } B(r))$ $\text{rel} \neq []$
 b) A $\text{rel} = []$
- Q3: $A \text{ AND ALL } r \text{ IN rel } (B(r))$
 $\Leftrightarrow$
 a) $\text{ALL } r \text{ IN rel } (A \text{ AND } B(r))$ $\text{rel} \neq []$
 b) A $\text{rel} = []$
- Q4: $A \text{ OR ALL } r \text{ IN rel } (B(r))$
 $\Leftrightarrow$
 $\text{ALL } r \text{ IN rel } (A \text{ OR } B(r))$
- Q5: $\text{SOME } r1 \text{ IN rel1 SOME } r2 \text{ IN rel2 } (A(r1, r2))$
 $\Leftrightarrow$
 $\text{SOME } r2 \text{ IN rel2 SOME } r1 \text{ IN rel1 } (A(r1, r2))$
- Q6: $\text{ALL } r1 \text{ IN rel1 ALL } r2 \text{ IN rel2 } (A(r1, r2))$
 $\Leftrightarrow$
 $\text{ALL } r2 \text{ IN rel2 ALL } r1 \text{ IN rel1 } (A(r1, r2))$
- Q7: $\text{SOME } r \text{ IN rel } (A(r) \text{ OR } B(r))$
 $\Leftrightarrow$
 $\text{SOME } r \text{ IN rel } (A(r)) \text{ OR SOME } r \text{ IN rel } (B(r))$
- Q8: $\text{ALL } r \text{ IN rel } (A(r) \text{ AND } B(r))$
 $\Leftrightarrow$
 $\text{ALL } r \text{ IN rel } (A(r)) \text{ AND ALL } r \text{ IN rel } (B(r))$
- Q9: $\text{NOT ALL } r \text{ IN rel } (A(r))$
 $\Leftrightarrow$
 $\text{SOME } r \text{ IN rel } (\text{NOT}(A(r)))$
- Q10: $\text{NOT SOME } r \text{ IN rel } (A(r))$
 $\Leftrightarrow$
 $\text{ALL } r \text{ IN rel } (\text{NOT}(A(r)))$

ცხრილი 1. კვანტორიანი გამოსახულების გარდაქმნის წესები

M1: Commulative rules		
a)	$A \text{ OR } B$	$\Leftrightarrow B \text{ OR } A$
b)	$A \text{ AND } B$	$\Leftrightarrow B \text{ AND } A$
M2: Associative rules		
a)	$(A \text{ OR } B) \text{ OR } C$	$\Leftrightarrow A \text{ OR } (B \text{ OR } C)$
b)	$(A \text{ AND } B) \text{ AND } C$	$\Leftrightarrow A \text{ AND } (B \text{ AND } C)$
M3: Distributive rules		
a)	$A \text{ OR } (B \text{ AND } C)$	$\Leftrightarrow (A \text{ OR } B) \text{ AND } (A \text{ OR } C)$
b)	$A \text{ AND } (B \text{ OR } C)$	$\Leftrightarrow (A \text{ AND } B) \text{ OR } (A \text{ AND } C)$
M4: Idempotency rules		
a)	$A \text{ OR } A$	$\Leftrightarrow A$
b)	$A \text{ AND } A$	$\Leftrightarrow A$
c)	$A \text{ OR } \text{NOT}(A)$	$\Leftrightarrow \text{TRUE}$
d)	$A \text{ AND } \text{NOT}(A)$	$\Leftrightarrow \text{FALSE}$
e)	$A \text{ AND } (A \text{ OR } B)$	$\Leftrightarrow A$
f)	$A \text{ OR } (A \text{ AND } B)$	$\Leftrightarrow A$
g)	$A \text{ OR } \text{FALSE}$	$\Leftrightarrow A$
h)	$A \text{ AND } \text{TRUE}$	$\Leftrightarrow A$
i)	$A \text{ OR } \text{TRUE}$	$\Leftrightarrow \text{TRUE}$
j)	$A \text{ AND } \text{FALSE}$	$\Leftrightarrow \text{FALSE}$
M5: De Morgan's rules		
a)	$\text{NOT } (A \text{ AND } B)$	$\Leftrightarrow \text{NOT } (A) \text{ OR } \text{NOT } (B)$
b)	$\text{NOT } (A \text{ OR } B)$	$\Leftrightarrow \text{NOT } (A) \text{ AND } \text{NOT } (B)$
M6: Double negation rules		
	$\text{NOT } (\text{NOT } (A))$	$\Leftrightarrow A$

ცხრილი 2. საერთო გამოსახულებებისათვის გარდაქმნის წესები

განსხვავებები განსაზღვრის არის დამოკიდებულებების ცარიელი და არა ცარიელი შემთხვევების დროს Q2 და Q3 წესებში ცხრილ 1-ში ჩნდება რელიაციური გამოთვლისას დროისა და მრავალ კლასობრიობას შორის კავშირის ცვალებადობის გამო [Jarke and Schmidt 1982]. რელიაციური გამოთვლის გამოსახულება შესაძლოა გარდაიქმნას ერთკლასიან გამოთვლად განსაზღვრის არის შემოღებით როგორიცაა ($r \text{ IN } \text{rel}$) კიდეც ერთი ატომარული პრედიკატის სახით:

```

O1: SOME r IN rel(pred)
<=> SOME r((r IN rel) AND pred),
O2: ALL r IN rel(pred)
<=> ALL r((r IN rel) T pred).

```

Q2a და Q3a წესების გამოყენება ტერმიდან კვანტორის გადადგილებისას განსაზღვრის არის ცარიელი დამოკიდებულების შემთხვევაში არასწორე რეზულტატს გამოიწვევდა. აქედან

გამომდინარეობს, რომ რელიგიური გამოთვლის გამოსახულების ნორმალიზაციისას, კომპილაციის დროს, აუცილებელია ცვლადების განსაზღვრის არეების საწყის განსაზღვრებაზე ინფორმაციის შენახვა რათა, აუცილებლობის შემთხვევაში, შესრულების დროს შესაძლებელი იყოს მოდიფიკაციის ჩატარება Q2b და Q3b წესების მიხედვით.

3.2 გამარტივება

უკვე აღინიშნა, რომ შესაძლოა ერთი და იმავე მოთხოვნის რამოდენიმე სემანტიკურად ექვივალენტური გამოსახულება არსებობდეს. ორ გამოსახულებას შორის ერთ-ერთ განსხვავების წყაროს შეიძლება სიჭარბის დონე წარმოადგენდეს [Hall 1976; Stroet and Engmann 1979]. ჭარბი გამოსახულების სწორხაზოვანი გამოთვლა მიგვიყვანდა ოპერაციების ნაკრების შესრულებამდე, რომელთაგან ზოგიერთი შესაძლოა ზედმეტი ყოფილიყო. ამიტომაც, მოთხოვნების ოპტიმიზატორი მიდრეკილია სიჭარბის აღმოფხვრისაკენ, ამ გამოსახულების გარდაქმნის გზით მის ექვივალენტურ გამოსახულებაში რომელიც ჭარბი არ იქნება.

ჭარბი გამოსახულება შეიძლება გამარტივებულ იქნას M4a-M4j გარდაქმნის წესების გამოყენებით, რომლებშიც გათვალისწინებულია იდეშპოტენტურობა (იხილეთ ცხრილი 2). ამ წესების გამოყენება რთულდება იმ ფაქტით, რომ იდეშპოტენტურობა საერთო ქვეგამოსახულებების არსებობის გამო შეიძლება გამოსახულების ნებისმიერ დონეზე შეგვხვდეს, ანუ იმ ქვეგამოსახულებების არსებობის გამო, რომელიც მოთხოვნის გამომხატველ გამოსახულებაში შეიძლება ერთ ჯერზე მეტად შეგვხვდეს. ამიტომაც, მაგალითად შემდეგი გამოსახულების გასამარტივებლად

```
[EACH e IN employees:
  e.ename = 'Smith'
  OR
  (e.status = assistant
   OR e.status = professor)
  AND
  NOT(e.status = professor
       OR e.status = assistant)]
```

ამ გამოსახულებამდე

```
[EACH e IN employees: e.ename = 'Smith']
```

M4d და M4g წესების გამოყენების საშუალებით, თავდაპირველად უნდა ამოვიცნოთ გამოსახულებების ექვივალენტურობა

$$(e.status = assistant \text{ OR } e.status = professor)$$

და

$$(e.status = professor \text{ OR } e.status = assistant)$$

ალგორითმები მოყვანილია [Downey et al. 1980 и Hall 1974, 1976]. საერთო ქვეგამოსახულებების გამოცნობა და იდემპოტენტურობის წესების გამოყენება უფრო ერთდროულად უნდა სრულდებოდეს, ვიდრე თანმიმდევრულად, რადგან იდემპოტენტურობის წესების მიხედვით გამოსახულებების გამარტივებისას შესაძლოა ახალი საერთო ქვეგამოსახულებები წარმოიშვას, რომლებიც, თავისმხრივ გასამარტივებელი იქნება. გამოსახულებები, რომლებიც ცარიელ კავშირებთან არიან დაკავშირებული შესაძლოა ასევე გამარტივდეს. მათი გამარტივების წესები ცხრილ 3-შია მოყვანილი (აღსანიშნავია, რომ ამ წესების გამოყენება მხოლოდ შესრულებისას შეიძლება)

E1:	[EACH r IN []: pred]	<==> []
E2:	[<r.A1,..., r.An> OF EACH r IN []: pred]	<==> []
E3:	SOME r IN [] (pred)	<==> FALSE
E4:	ALL r IN [] (pred)	<==> TRUE

ცხრილი 3. ცარიელი კავშირების მეორე გამოსახულებების გარდაქმნის წესები

ტერმები, როგორც განსაზღვრულია 2.1 თავში რელიაციურ გამოთვლაში ატომარულ პრედიკატებს წარმოადგენენ. თუმცა, ეს ტერმები შეიძლება გამარტივდეს ან წაიშალოს, თუ შედარების ოპერაციის სემანტიკა აშკარადაა გათვალისწინებული. მნიშვნელოვან დანართს წარმოადგენს ეგრეთ წოდებული კონსტანტების გავრცელება, რომელშიც ტრანზიტულობის კანონი გამოიყენება, როგორიცაა

$$r.A \text{ op } s.B \text{ AND } s.B = const$$

$$\Rightarrow r.A \text{ op } const,$$

რომლებიც მოთხოვნაში ორ ადგილიანი ტერმების რაოდენობის შემცირების საშუალებებს იძლევა. ტაბლოში სტრიქონების რაოდენობის მინიმიზაციის ალგორითმებში (თავი 2.4)

სისტემატურად გამოიყენება ასეთი გამარტივების წესები კონიუნქტიური მოთხოვნებისათვის. რადგანაც ტაბლოში სტრიქონების რაოდენობა მეტია გამოსახულებაში კავშირების რაოდენობაზე (ორადგილიანი მაკავშირებელი ტერმების), სტრიქონების რაოდენობის მინიმიზაცია შეესაბამება ჭარბი (ზედმეტი) კავშირების აღმოფხვრას.

საგივა და იანაკაკისმა [Sagiv and Yannakakis 1980] გააფართოვეს ტაბლოს მეთოდები დიზუნქტივების შემცველი გამოსახულებების გამარტივების უზრუნველსაყოფად. თუმცა, მეთოდების განზოგადოება ყველა რელიაციურად სრული ენის გამოსახულებებისათვის კვლავ გადაუწყვეტ პრობლემად რჩება.

მოთხოვნების სემანტიკური გამარტივებისათვის შესაძლოა გამოვიყენოთ ინფორმაცია მთლიანობის სემანტიკურ შეზღუდვებზე. მარტივი მაგალითის სახით განვიხილოთ გასაღების შეზღუდვის მაგალითი. თუ r და r' წარმოადგენენ (არსებობის კვანტორის მიერ თავისუფალ ან შეზღუდულ) ცვლადებს ერთი და იმავე განსაზღვრის არით, "rel" დამოკიდებულებით, მაშინ " $r.key = r'.key$ " სახის ექვიკავშირი წარმოადგენს ჭარბს, იმ გაგებით, რომ ეს ტერმი და ერთ-ერთი ცვლადი მაგალითად r' შესაძლოა წაიშალოს r -ის ჩანაცვლებით ყველა იმ ტერმში, რომელშიც r' -შედის. გამარტივების ეს ტიპი განსაკუთრებით ხელსაყრელია გამოსახულების გამოთვლის კონტექსტში, როდესაც მომხმარებლების მოთხოვნების გარდაქმნამ სისტემურ მოთხოვნებად შენახული დამოკიდებულებებით შეიძლება გარკვეული სიჭარბე გამოიწვიოს. სემანტიკური გამარტივების გამოყენების არე შეიძლება გაფართოვდეს მოთხოვნის სტრუქტურით გამოწვეული დამატებითი შეზღუდვების დაწესებით.

გამარტივების საბოლოო შესაძლებლობა იმ შემთხვევებში ჩნდება, როდესაც იმის ცვენება გამოდის, რომ სტანდარტიზირებული მოთხოვნის მატრიცის ერთი ან რამდენიმე კონიუნქტი არასოდეს დაკმაყოფილდება. მაგალითად:

$$r.A \geq s.B \text{ AND } s.B > t.C \text{ AND } t.C \leq r.A,$$

$r.a > r.a$ წინააღმდეგობას ქმნის და ამიტომაც ბულის ტიპის მნიშვნელობით false შეიძლება იქნას ჩანაცვლებული. ტერმის პირობის შეუსრულებლობის პრობლემა ეფექტურად იჭრება კომპილაციის დროს იმ ტერმების კონიუნქციისათვის სადაც ტოლობის ოპერაცია ხდება ($=$, $<$, $\<e$, $>$, $\>e$) თუმცა, გამოსათვლელად რთულია თუ უთანასწორობაზე შედარება სრულდება.

3.3 გაუმჯობესება

მოთხოვნის გამარტივების რეზულტატად არ არის აუცილებელი, რომ უნიკალური გამოსახულება მივიღოთ. შესაძლოა, არსებობდეს სხვა არა ჭარბი გამოსახულებები რომლებიც სემანტიკურად ექვივალენტური იქნება რომელიმე გამარტივების მეთოდის მიერ გენერირებული გამოსახულებების. გამოსახულებების გამოთვლა, რომელიც ერთი და იმავე მოთხოვნას შეესაბამებინ, შესაძლოა არსებითად განსხვავდებოდეს ეფექტურობის პარამეტრებით, მაგალითად შუალედური რეზულტატების ზომით და დამოკიდებულების ელემენტებთან მიმართვის რაოდენობით. ქვემოთ მოყვანილია მთელი რიგი ევრისტიკა, მოთხოვნების ისეთი გარდაქმნებისათვის რომელთა შესრულებაც გამოსახულების შესრულების ეფექტურობას აუმჯობესებს.

უმარტივეს გარდაქმნებს, რომელიც ამ თავში განიხილება, წარმოადგენს პროექციების მიმდევრობის ერთ პროექციად გაერთიანება და შეზღუდვების მიმდევრობის ერთ შეზღუდვად გაერთიანება. შესაბამისი გარდაქმნის წესებია:

```
A1: Proj( . . . Proj( Proj( rel, A1), A2), . . . , An)  
==> Proj( rel, An),  
A2: Rest( . . . (Rest( Rest( rel, pred1), pred2) . . . , predn)  
==> Rest( rel, pred1 AND pred2 AND . . . AND predn).
```

დამოკიდებულებებს შორისი ოპერაციების გაერთიანება ორ უპირატესობას უზრუნველყოფს. პირველია, რომ ერთი და იმავე კავშირის რამოდენიმეჯერ წაკითხვა აღმოიფხვრება, ხოლო მეორე კი არსებული წვდომის გზები შესაძლოა გამოყენებულ იქნას გაერთიანებული ოპერაციისათვის და არა მხოლოდ მიმდევრობის პირველი ოპერაციისათვის.

გაუმჯობესების გარდაქმნების მიზანს წარმოადგენს შუალედური რეზულტატების კონსტრუირების, შენახვის და გამოთვლის ზომის მინიმიზაცია. მნიშვნელოვანი ევრისტიკა გადაადგილებს შერჩევის ოპერაციებს, როგორებიცაა შეზღუდვა და პროექცია, კონსტრუქციურ ოპერაციებამდე, როგორებიცაა გაერთიანება და დეკარტული ნამრავლი, რათა შერჩევის ოპერაციები რაც შეიძლება ადრე შეასრულოს. რელიაციური გამოთვლის კონტექსტში გამოთვლების გარკვეული თანმიმდევრობის დაშლა შესაძლოა წარმოვადგინოთ როგორც ჩადგმული გამოსახულება. ჩადგმული გამოსახულების გამოთვლა იწყება ყველაზე შიდა ჩადგმული გამოსახულების გამოთვლით რომლის შემდგეგაც უკვე მისი უშუალოდ მომდევნო

გამოსახულების შესრულება სრულდება და ა.შ. ჩადგმული გამოსახულება, რომელიც ერთ ადგილიანი ტერმების ადრეულ გამოთვლას გულისხმობს მოყვანილია 3.1 მაგალითში.

მაგალითი 3.1. ჩადგმული გამოსახულება, რომელიც ექვივალენტურია 2.1 მაგალითში მოყვნილი გამოსახულების

```
[<e.ename) OF  
EACH e IN [EACH e IN employees:  
    e.status = professor]:  
SOME p IN [EACH p IN papers:  
    p.year = 1981]  
(e.enr = p.enr)]
```

შერჩევის ოპერაციების ადრეული გამოთვლები წარმოადგენს მოთხოვნის გამოყოფის (query detachment) კერძო შემთხვევას. ქვეგამოსახულება, რომელიც მოთხოვნის დარჩენილ ნაწილთან ერთი ცვლადით გადაიფარება, გამოიყოფა და ქმნის შიდა ჩადგმას. გამოყოფა სრულდება ჩადგმის ყოველ დონეზე რეკურსიულად მანამდე, სანამ გამოსახულება არ შეწყვეტს შემცირებას. ექსპერიმენტებმა აჩვენეს, რომ ეს ძალიან ძლიერი ევრისტიკაა. 3.2 მაგალითში დემონსტრირებულია რთული გამოსახულებისაგან ქვეგამოსახულების გამოყოფა.

მაგალითი 3.2. განყოფილებები, რომლებიც სთავაზობენ ლექციებს, რომლებსაც ატარებენ პროფესორები, რომლებიც იმავე ქალაქში ცხოვრობენ სადაც მდებარეობს განყოფილება და რომელთაგან თიეთოეულს გამოქვეყნებული აქვს სტატია 1981 წელს.

შესაბამისი გამოსახულება:

```
[EACH d IN departments:  
    SOME l IN lectures  
    SOME e IN employees  
        (e.status = professor  
        AND  
        d.dname = l.dname  
        AND l.enr = e.enr  
        AND e.city = d.city  
        AND  
        SOME p IN papers
```

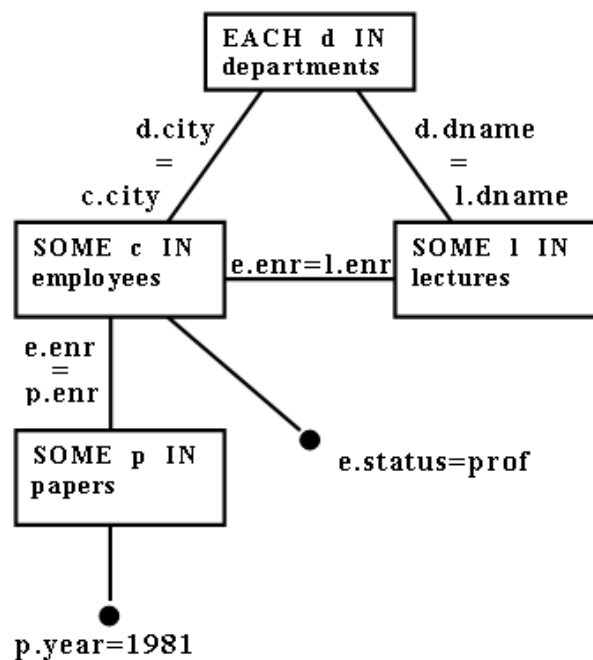
```
(p.year = 1981
AND p.enr = e.enr))]
```

ექვივალენტური გამსახულება, რომელიც სრულდება მოთხოვნის გამოყოფით:

```
[EACH d IN departments:
  SOME l IN lectures
    SOME e IN [EACH e IN
      [EACH e IN employees:
        e.status = professor]:
      SOME p IN [EACH p IN
        papers: p.year = 1981]
      (e.enr = p.enr)]
    (d.dname = l.dname AND l.enr = e.enr
    AND e.city = d.city)]
```

ობიექტური გრაფი რომელიც მოთხოვნას წარმოადგენს, ნაჩვენებია ნახაზ 4-ზე.

აღსანიშნავია, რომ შედეგად მიღებული ჩადგმული გამოსახულება განუყოფელია ანუ მისი დაყოფა ორ ქვეგამოსახულებად რომელიც ერთი ცვლადით გადაიფარება არ შეიძლება. სხვა სიტყვებით, ჩადგმული გამოსახულება ციკლს შეიცავს.



ნახაზი 4. ობიექტური გრაფი 3.2 მაგალითისთვის

აქამდე მოყვანილ გაუმჯობესების გარდაქმნებში გამოიყენება ინფორმაცია სამი წყაროდან: გარდაქმნების და ევრისტიკის საერთო წესები, რომელიც მათ გამოყენებას მართავს, რელიაციური მონაცემების სტრუქტურის ცოდნა და თვითონ მოთხოვნა. ჯერ არ განგვიხილავს ცოდნის ორი სხვა წყარო: მთლიანობის შეზღუდვები, რომლებიც მონაცემთა ბევრ ბაზაში სქემის სტრუქტურულ განსაზღვრებას ავსებენ და მონაცემთა ბაზაში არსებული რეალური მონაცემები.

მთლიანობის შეზღუდვა- წარმოადგენს პრედიკატებს, რომლებიც ჭეშმარიტი უნდა იყოს გარკვეული დამოკიდებულების ყოველი ელემენტისათვის ან გარკვეული დამოკიდებულებების ჯგუფის ელემენტების ყოველი კომბინაციისათვის. ამიტომაც მათი დამატება შესაძლებელია ნებისმიერი მოთხოვნის შერჩევის გამოსახულებისათვის მისი ჭეშმარიტი მნიშვნელობის შეცვლის გარეშე. არსებობს რამოდენიმე მიდგომა, რომლებშიც ეს დაკვირვება უფრო გასაუმჯობესებლად ვიდრე გასამარტივებლად გამოიყენება. მათ მოთხოვნის ცოდნაზე დაფუძნებული (knowledge-based) სემანტიკური დამუშავება ეწოდებათ.

დავუშვათ, მაგალითად, რომ მთლიანობის შეზღუდვა შემდეგია: "ჩვენ სამუშაოზე ვიყვანთ მხოლოდ იმ პროფესორებს რომლებიც წელიწადში მინიმუმ ერთ სტატიას აქვეყნებენ". ამ შემთხვევაში 2.1 მაგალითის (რომელშიც გვანტერესებდა პროფესორების სახელები რომელთაც სტატიები 1981 წელს გამოაქვეყნეს) მოთხოვნის გამოთვლა ტრივიალური ხდება, ხოლო 3.2 მაგალითის მოთხოვნის გამოთვლა არსებითად მარტივდება.

მთლიანობის შეზღუდვის დამატებას შერჩევის გამოსახულებაზე შეუძლია ასევე შეცვალოს მოთხოვნის სტრუქტურა, გახდის რა მას მეტად შესაფერისს დამუშავებისათვის. თუ, დავამატებთ შეზღუდვას "ჩვენ ვიყვანთ მხოლოდ ადგილობრივ პროფესორებს" ამ შემთხვევაში 3.2 მაგალითში ტერმი "d.city = e.city" შეიძლება დაიწიოს და დაჩენილი მოთხოვნის ობიექტურ გრაფში ციკლი აღარ იქნება.

სემანტიკური დამუშავების წარმატება მნიშვნელოვნად არის დამოკიდებული ბევრ გარდაქმნებს შორის არჩევანის გასაკეთებლად საჭირო ეფექტური ევრისტიკის შემუშავებაზე, რომლებიც მოთხოვნისათვის მთლიანობის შეზღუდვის ნებისმიერი კომბინაციის დამატების საშუალებას იძლევიან.

არსებობს შემთხვევები, როდესაც ოპტიმალური გარდაქმნა მონაცემებზეა დამოკიდებული. ზემოთ მოყვანილი ევრისტიკები შესაძლოა ყოველთვის არ იყოს ოპტიმალური განსაკუთრებით, იმ შემთხვევებში, როდესაც ზოგიერთი წვდომის გზა მხარდაჭერილია ფიზიკური შენახვის სტრუქტურებით. თუმცა, მონაცემებზე ასეთი დამოკიდებულების შედეგია ის, რომ მოთხოვნის გარდაქმნის საშუალებები მხარდაჭერილი უნდა იყოს არა მხოლოდ კომპილაციის არამედ შესრულების დროსაც. გარდა ამისა, თუ ევრისტიკებს არ მოაქვთ დამაკმაყოფილებელი შედეგი, საჭიროა ერთდროული ოპტიმიზაცია ფიზიკურ და ლოგიკურ დონეებზე. თუმცა ასეთ ინტეგრირებულ მიდგომებზე გადასვლამდე საჭიროა მოთხოვნის კომპონენტების შესრულების ფიზიკური აღწერა.

4. მოთხოვნების გამოთვლა

ამ თავში წარმოდგენილი იქნება სსხვადასხვა სირთულის მოთხოვნის კომპონენტების გამოთვლის მეთოდები, როგორცაა გამოსახულება ერთადერთი ცვლადით, გამოსახულება ორი ცვლადითა და გამოსახულება ბევრი ცვლადით. ინდივიდუალური მიდგომები შეიძლება წარმოვიდგინოთ როგორც მოთხოვნების გამოთვლის საერთო სისტემის სამშენებლო ბლოკები. მათი ღირებულება და გამოყენების დიაპაზონი ქმნის საბოლოო მონაცემებს მოთხოვნების ოპტიმიზაციის უკანასკნელი ეტაპისათვის, რომელშიც წვდომის ოპტიმალური გეგმა გენერირდება.

4.1 გამოსახულება ერთი ცვლადით

გამოსახულებები ერთი ცვლადით აღწერენ ელემენტების შერჩევის პირობას ერთი დამოკიდებულებიდან. მათი გამოთვლის „გულუპრყვილო“ მეთოდი მდგომარეობს დამოკიდებულების თითოეული ელემენტის წაკითხვაში და იმის შემოწმებაში აკმაყოფილებს ის თუ არა გამოსახულების ყოველ ტერმს. რადგანაც ეს მიდგომა მსხვილი დამოკიდებულებებისა და რთული გამოსახულებების არსებობის დროს ძალიან ძვირადღირებულია, ელემენტებთან მიმართვის და წაკითხული ელემენტების შემოწმების რაოდენობის შესამცირებლად განსხვავებული მეთოდების გამოყენება ხდება.

პირდაპირი და დალაგებული (თანმიმდევრული) წვდომა შესაძლოა უზრუნველყოფილ იქნას ინდექსებით, რომელთა გაერთიანებაც შეიძლება მულტიცხრილურ სტრუქტურებთან [Welch and Graham 1976; Yang 1977]. კონცეპტუალური თვალსაზრისით, ინდექსი არის ბინარული

დამოკიდებულება, რომელიც ატრიბუტების მნიშვნელობებს აკავშირებს დამოკიდებულებების ელემენტებზე მიმართვებთან, რომელთაც ჩვეულებრივ კორტეჟების იდენტიფიკატორებს უწოდებენ (TID). მოვახდინოთ ერთგანზომილებიანი ინდექსების განვასხვავება მრავალგანზომილებიანი ინდექსებისაგან, პირველი წვდომას უზრუნველყოფს დამოკიდებულების ერთი ატრიბუტის მეშვეობით, ხოლო მეორე კი წვდომას ატრიბუტების კომპინაციით ასრულებს. ერთგანზომილებიანი ინდექსების რეალიზება ჩვეულებრივ ISAM [IBM 1966] სტრუქტურის ან B-ხეების [Bayer and McCreight 1972] საფუძველზე ხორციელდება. მრავალგანზომილებიანი ინდექსების სტრუქტურა აღწერილია [Bentley and Friedman 1979].

შემოწმებების რაოდენობა, რომელიც გამოსახულების გამოთვლისას დამოკიდებულების შერჩეულ ელემენტზე გამოიყენება, შესაძლოა შემცირდეს შესრულების დროს გამოსახულების გარდაქმნის გზით. გამოსახულებების სპეციალური კლასის, ბულის გამოსახულებების, ოპტიმიზაცია დიდი ხანია რაც კვლევის საგანს წარმოადგენს კომპილატორების კონსტრუირების სფეროში. ბულის გამოსახულებები (ტერმები რომელთაც არ აქვთ კვანტორები და რომლებიც AND/OR-ით არიან დაკავშირებული) წარმოადგენს მაღალი დონის პროგრამირების ენების მთელ რიგ მართვის სტრუქტურების შემადგენელ ნაწილს. ბულის ტიპის გამოსახულებების ოპტიმიზაციის მიზანი მდგომარეობს კოდის გენერაციაში, რომელიც აღარ ითვლის გამოსახულებების იმ კომპონენტებს, რომლებიც მთლიანად გამოსახულების მნიშვნელობისათვის აღარ არის არსებითი. მაგალითად, შემდეგ ოპერატორში:

```
IF A AND B THEN
    statement_1
ELSE
    statement_2
END,
```

თუ ტერმი A უკვე გამოთვლილია და "false" მნიშვნელობა აქვს, მაშინ ტერმ B-ს გამოთვლა ზედმეტი იქნება და შეიძლება პირდაპირ ELSE- შესრულდეს. თუ იგივე იდეა გამოყენებულ იქნება ერთგვალადიანი გამოსახულებების გამოთვლისას [Gudes and Reiter 1973; Liu 1976] მაშინ მოთხოვნების გამარტივება უშუალოდ შესრულებისას იქნება შესაძლებელი.

სხვა მიდგომა, რომელიც შემუშავებულ იქნა ეფექტურობის სრულყოფისათვის, მდომარეობს გამოსახულების ინდივიდუალური კომპონენტების გამოთვლის რიგითობის შეცვლაში. ცნობილია, რამოდენიმე ალგორითმი, რომელიც გარკვეულ შემთხვევებში შესრულების

ოპტიმალურ თანმიმდევრობას აღწევს. ზოგიერთ მათგანში ატრიბუტების მნიშვნელობების აპრიორული ალბათობის არსებობა ივარაუდება [Hanani 1977], ხოლო სხვები კი ამ მოსაზრებების გარეშე ახორციელებენ მუშაობას [Breitbart and Reiter 1975].

4.2 გამოსახულებები ორი ცვლადით

გამოსახულებები ორი ცვლადით აღწერენ პირობას ორი დამოკიდებულებიდან ელემენტების კომბინაციისათვის. საერთოდ, ორცვლადიანი გამოსახულება შედგება ერთადგილიანი ტერმებისაგან, რომლებიც ცალკეულ ცვლადებს ერთმანეთისგან დამოუკიდებლად ზღუდავენ და ორადგილიანი ტერმებისაგან, რომლებიც ამ ორივე ცვლადს შორის კავშირს ამყარებენ.

გაერთიანების ოპერაციის რეალიზაციის მიდგომები შეიძლება კლასიფიცირდეს სტრატეგიებად, რომლებიც რიგითობაზეა დამოკიდებული და სტრატეგიებად რომლებიც არ არის რიგითობაზე დამოკიდებული [Todd 1974]. მარტივი მეთოდი, რომელიც არ არის დამოკიდებული ელემენტებთან წვდომის რიგითობაზე ჩადგმული ინტეგრაციის მეთოდია (nested iteration) [Pecherer 1975, 1976; Selinger et al. 1979] რომელშიც დამოკიდებულების ელემენტების თითოეული წყვილი ითვლება და თუ კავშირის პირობები კმაყოფილდება ეს წყვილი კონკატენაციას განიცდის. ალგორითმის მონახაზი შემდეგნაირად გამოიყურება:

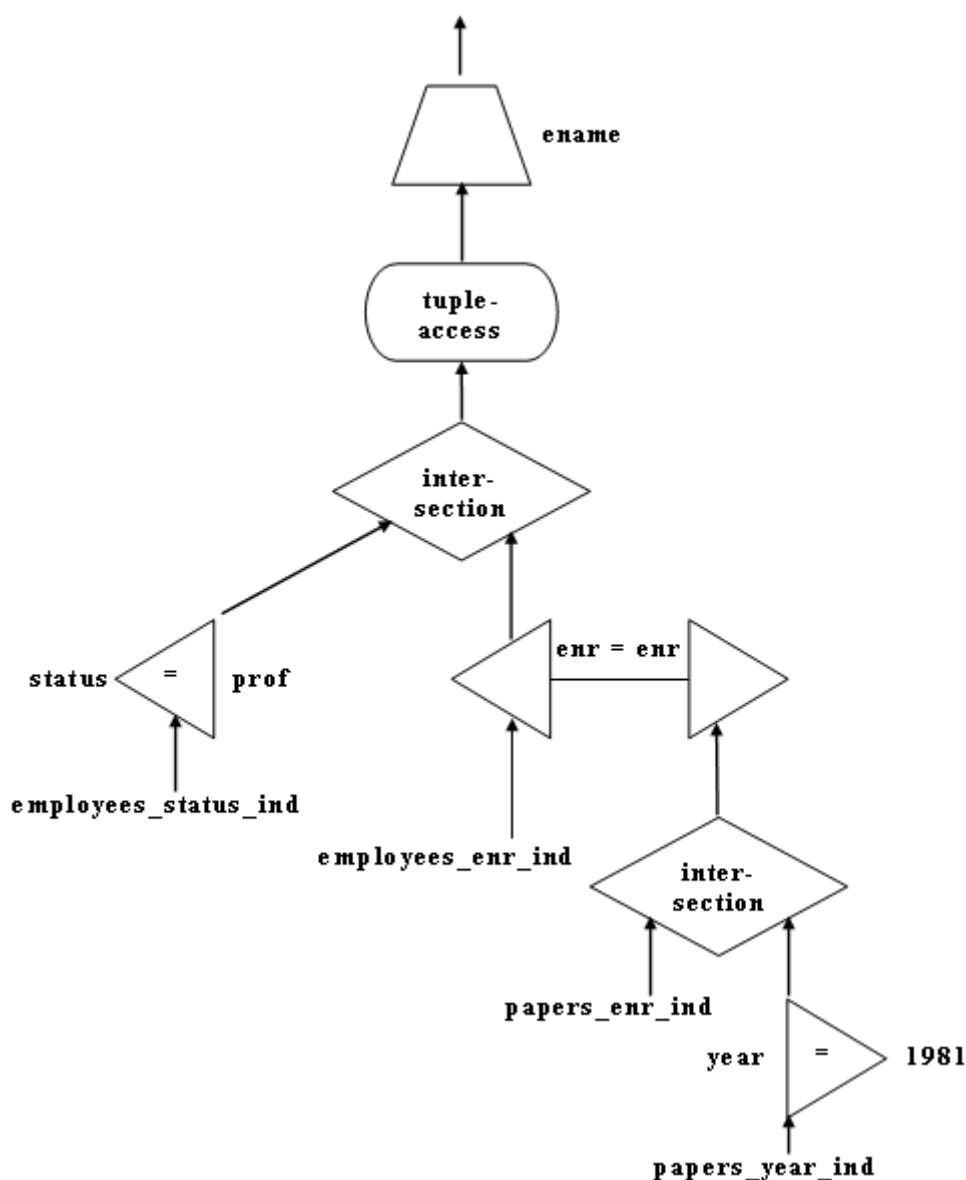
```
FOR i := 1 TO N1 DO
  read ith element of rel1;
  FOR j := 1 TO N2 DO
    read jth element of rel2;
    form the join according to the join condition;
  END;
END;
```

დავუშვათ, რომ $N_1(N_2)$ -კავშირის ელემენტების რაოდენობაა, რომელიც შიდა ციკლში ითვლება. ორადგილიანი ტერმის გამოსათვლელად საჭიროა $N_1 + N_1 * N_2$ მეორად მეხიერებასთან მიმართვა, თუ ჩავთვლით, რომ ყოველი ელემენტისათვის მეორად მეხიერებასთან მიმართვაა საჭირო.

ჩადგმულ იტერაციის მეთოდს შესაძლოა დაემატოს "rel₂" დამოკიდებულების ატრიბუტზე(ებზე) ინდექსის გამოყენება. "rel₁"-ის ყოველი ელემენტისათვის, "rel₂"-ის თითოეული ელემენტის თანმიმდევრული სკანირების მაგივრად, "rel₂"-ის შეწყვილებული

ელემენტები პირდაპირ შერჩევა [Griffeth 1978; Klug Klug 1982b]. ამგვარად საჭირო იქნება მხოლოდ $N_1 + N_1 * N_2 * j_{12}$ მიმართვა სადაც j_{12} -შერჩევის კოეფიციენტი რომელიც ახასიათებს "rel₁" და "rel₂" დეკარტული ნამრავლის შემცირებას კავშირის პირობასთან შესაბამისად.

შერწყმის მეთოდი (merge method) [Blasgen and Eswaran 1977; Selinger et al. 1979] დაფუძნებულია რიგითობაზე, რომელშიც სრულდება დამოკიდებულების ელემენტებთან წვდომა. ორივე დამოკიდებულება კავშირის ატრიბუტების მნიშვნელობების მიხედვით ზრდადობით ან კლებადობით სორტირდება და კავშირის პირობის შესაბამისად შერწყმა ხორციელდება. მეორად მეხსიერებასთან დაახლოებით $N_1 + N_2 + S_1 + S_2$ მიმართვაა საჭირო, სადაც S_1 და S_2 ორივე დამოკიდებულების სორტირებისათვის მეორად მეხსიერებასთან მიმართვის საჭირო რაოდენობას აღნიშნავს. თუ ორივე დამოკიდებულება უკვე დასორტირებულია ამავე კრიტერიის მიხედვით, მაშინ შერწყმის მეთოდი ორადგილიანი ტერმების გამოთვლის ყველაზე მეტად ეფეტურ მეთოდად მიიჩნევა [Merrett 1981; Merrett et al. 1981]. გამონაკლისს წარმოადგენს ის შემთხვევები როდესაც ორივე დამოკიდებულება საკმაოდ მცირეა იმისათვის რომ დაეტიოს ძირითად მესიერებაში (ამ დროს უკეთესია ჩადგმული ბლოკების მეთოდის გამოყენება) და როდესაც ერთი დამოკიდებულება მეორეზე საკმაოდ დიდია და შესაძლებელია ინდექსების გამოყენება (ამ შემთხვევაში უკეთესად იმუშავებს ჩადგმული იტერაცია ინდექსების გამოყენებით).



ნახაზი 5. ოპერაციის გრაფი, რომელიც 2.1 მაგალითში მოყვანილ მოთხოვნის გამოთვლის ილუსტრირებას ახდენს. ივარაუდება განსხვავებული ინდექსების არსებობა

ორცვლადიანი შემთხვევითი გამოსახულების (arbitrary expression) გამოთვლის მეთოდები იქმნება ერთცვლადიანი გამოსახულების სტრატეგიის საფუძველზე და ორადგილიანი ტერმების გამოთვლის ალგორითმით. ისინი განსხვავდებიან, როგორც წვდომის გზების, როგორებიცაა ინდექსი და სორტირება, გამოყენების და შექმნის მეთოდებით ასევე ტერმის დამუშავების რიგითობითაც. ერთ-ერთი ასეთი მეთოდი ილუსტრირებულია ნახაზ 5-ზე. მასში ფართოდ გამოიყენება ინდექსები. კორტეჟების იდენტიფიკატორები, რომლებიც ერთადგილიანი ტერმის დამუშავებით მიიღება და ისინი რომლებიც აკმაყოფილებენ დაკავშირების პირობას გადაიკვეთება და შემგედ, დამოკიდებულების ელემენტებთან წვდომისათვის გამოიყენება. ეს

ელემენტები პროეცირდება ატრიბუტებზე რომლებიც ორადგილიან ტერმებში და სამიზნე სიაში გვხვდება. პროეცირებადი ელემენტები კონკატენაციას განიცდიან და საბოლოოდ პროეცირდებიან სამიზნე ატრიბუტზე.

[Blasgen and Eswaran 1976, Niebuhr and Smith 1976, Yao and DeJong 1978 и 1979] აღწერილია ბევრი განსხვავებული ალგორითმი და მოყვანილია მათი სისტემური შედარება ეფექტურობის მიხედვით. ეს შედეგები დემონსტრირებას ახდენენ იმისა, რომ არ არსებობს აპრიორი საუკეთესო ალგორითმი. ოპტიმიზატორი ან უნდა დაეყრდნოს ვრისტიკას ან მან თითოეული მოთხოვნისათვის უნდა ჩაატაროს ბევრი ალტერნატივის ღირებულების შედარებები, რაც თავისთავად ძვირადღირებულია.

4.3 გამოსახულებები რამდენიმე ცვლადით

რამდენიმე ცვლადიანი გამოსახულების გამოთვლის სტრატეგიები მოთხოვნის დამუშავების ზოგადი სისტემის მშენებლობისათვის საჭირო ყველაზე დიდ ბლოკებს წარმოადგენენ. არსებობს ორი ძირითადი მიდგომა, რომელთაც პარალელური დამუშავება და ეტაპობრივი შემცირება (stepwise reduction) ეწოდებათ.

მოთხოვნის კომპონენტების პარალელური დამუშავება ემსახურება იმას, რომ ერთიდაიმავე მონაცემთან განმეორებადი მიმართვები თავიდან იქნას აცილებული. განმეორებადი მიმართვები ერთიდაიმავე მონაცემთან შეიძლება თავიდან ავიცილოთ მოთხოვნის რამდენიმე კომპონენტის ერთდროული გამოთვლით. [Palermo 1972]-ში ყველა ერთადგილიანი ტერმი, რომელიც გარკვეულ ცვლადთან არის დაკავშირებული, სრულად გამოითვლება და ყველა ორადგილიანი ტერმი კი, რომელშიც იგივე ცვლადი გვხვდება ნაწილობრივ მუშავდება ეს ყველაფერი ერთდროულად სრულდება, ამ ცვლადის მიერ განსაზღვრული დამოკიდებულების სფეროს სკანირების გზით. პარალელურად შეიძლება გამოვითვალოთ ტერმებს შორის AND კავშირებიც, რაც უფრო მეტად ამცირებს შუალედური რეზულტატების ზომას [Jarke and Schmidt 1982]. ანალოგიური მიდგომა უფრო მაღალ დონეზე აღწერილია [Klug 1982b]. ამ მიდგომაში პარალელურად გამოითვლება აგრეგატული ფუნქციები და რთული ქვემოთხოვნები. მოთხოვნის კომპონენტების პარალელური დამუშავებისათვის სტრატეგიების დაგეგმვა ასევე აღწერილია შმიდტის მიერ [Schmidt 1979].

სხვა მიდგომას, რომელიც პარალელური დამუშავების უპირატესობებს გამოიყენებს, წარმოადგენს კონვეიერული ოპერაციები, რომელთაც შეუძლიათ მუშაობა წინა ოპერაციების

ნაწილობრივ შედეგებთან. მაგალითად, შეზღუდვა და პროექცია შესაძლოა კონვეიერებული იქნას, ისე რომ მონაცემთა გაცვლისათვის მხოლოდ შედარებით მცირე ბუფერი იქნება საჭირო.

ერთდროული გამოთვლის და კონვეირიზაციის ასპექტები ერთიანდება ეგრეთწოდებულ "უკუ კავშირში" ("feedback" method) [Clausen 1980; Rothnie 1974, 1975], რომელშიც გაერთიანების ოპერაციის ნაწილობრივი შედეგები გამოიყენება მისი შემავალი მონაცემების შეზღუდვისათვის. ის თუ რა დონით შეიძლება ამის გაკეთება, დამაკავშირებელ ტერმში შემავალი მონაცემების კვანტიფიკაციაზეა დამოკიდებულია. განვიხილოთ გამოსახულების მაგალითი:

$$[EACH\ r_1\ IN\ rel_1:\ ALL\ rel_2\ IN\ rel_2\ (r_1.A\ op\ rel_2.B)].$$

დავუშვათ, რომ დამაკავშირებელი ტერმი გამოითვლება ჩადგმული იტერაციის მეთოდით და გარკვეული r_1 ელემენტის შემოწმებისას აღმოჩნდა რომ ტერმი " $r_1.A\ op\ rel_2.B$ " გვაძლევს false-ს გარკვეული $rel_2\ c\ rel_2.B = c_1$. რადგანაც rel_2 არსებობის კვანტორითაა დაკავშირებული, r_1 - უარყოფთ და შერჩევის გამოსახულებაში შესაძლებელია გამომრიცხავი ფილტრის დამატება

$$[EACH\ r_1\ IN\ rel_1:\ NOT\ (r_1.A\ op\ c_1) AND\ ALL\ rel_2\ IN\ rel_2\ (r_1.A\ op\ rel_2.B)],$$

რადგანაც rel_2 -ის ერთი და იგივე მნიშვნელობა გამოიწვევდა ყველა r_1 ელემენტის რომელიც არ აკმაყოფილებენ პირველ ტერმს გადახრას " rel_2 "-დან. მეორე მხრივ, თუ $r_1\ c\ rel_1.A = c_2$ გადის შემოწმებას შერჩევის პრედიკატისათვის "ჭეშმარიტების" ფილტრის დამატება შეიძლება:

$$[EACH\ r_1\ IN\ rel_1:\ r_1.A\ op\ c_2\ OR\ NOT\ (r_1.A\ op\ c_1)\ AND\ \dots].$$

შეზღუდვის გასაძლიერებლად შემდგომში ორივე ფილტრის განახლებაა შესაძლებელი.

რამოდენიმე ცვლადიანი გამოსახულების გამოთვლის მეორე მიდგომა დასაბუთებულია შემდეგი მაგალითის მიხედვით:

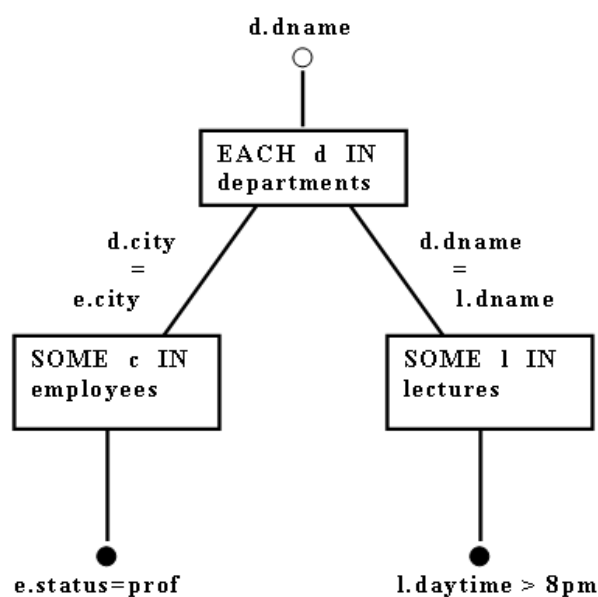
მაგალითი 4.1. ნახაზ 6-ზე ნაჩვენებია ობიექტური გრაფი რომელიც გამოსახულებას გამოხატავს

```
[<d.dname> OF
  EACH d IN departments:
    SOME e IN employees(e.status = professor
                        AND
                        e.city = d.city)

  AND

  SOME l IN lectures (l.daytime > 8 pm
                    AND
                    l.dname = d.dname)]
```

4.1 მაგალითში მოყვანილი გამოსახულების მსგავს გამოსახულებებს ხისებრი გამოსახულებები ეწოდება, რადგანაც მათთან ასოცირებული გრაფი ხეს წარმოადგენს. ასეთი გამოსახულების გამოთვლის სტანდარტული მიდგომა შედგება სამივე დამოკიდებულების გაერთიანების ფორმირებით, ერთ ადგილიანი ტერმებით შუალედური შედეგის შეზღუდვით და სამიზნე სიაში მითითებულ ატრიბუტებზე საბოლოო პროექციით. როგორც შემავალ მაგალითშია ნაჩვენები ეს მეთოდი საკმაოდ ცუდად მუშაობს. ეს უფრო მეტად პრობლემატურია დანაწილებულ გარემოში, სადაც თითოეული დამოკიდებულება თავის კვანძშია განლაგებული და საჭიროა მთლიანი დამოკიდებულებების გადაცემა კვანძიდან-კვანძში. უფრო მეტიც, დამოკიდებულება სამიზნე კვანძში შესაძლოა დროებით გაფართოვდეს კავსირის ფორმირებისათვის თუმცა, საბოლოო რეზულტატი წარმოადგენს მხოლოდ მის ჰორიზონტალურ და ვერტიკალურ ქვესიმრავლეს.



ნახაზი 6. მაგალითის ობიექტური გრაფი

[Bernstein and Chiu 1981, Bernstein et al. 1981 и Bernstein and Goodman 1981a, 1981b, 1981c]-ში შემოთავაზებულია ხისებრი გამოსახულებების ეტაპობრივი შემცირება (თავისუფალი და არსებობის კვანტორით დაკავშირებული ცვლადებით). ეს მეთოდი წარმადობით ხშირად აღემატება ზემოთ აღწერილ უბრალო მიდგომას, როგორც დეცენტრალიზებულ ასევე ცენტრალიზებულ გარემოში. ის ეფუძნება დაკავშირების მოდიფიცირებულ ოპერაციას და მასში გამოიყენება გრეთ წოდებული ნახევრად დაკავშირების ოპერაცია.

" rel_1 " დამოკიდებულების ნახევრად დაკავშირება " rel_2 "-თან უტოლდება ამ დამოკიდებულებების დაკავშირებას, სპროეცირებადს " rel_1 " დამოკიდებულების ატრიბუტებზე:

```
semijoin( $rel_1$ ,  $A \text{ op } B$ ,  $rel_2$ )  
= proj(join( $rel_1$ ,  $A \text{ op } B$ ,  $rel_2$ ), attributes  $rel_1$ )).
```

ამგვარად, ეს ოპერაცია ქმნის "ნახევარ კავშირს". ნახევარ კავშირის ძირითადი უპირატესობები შემდეგში მდგომარეობს: (1) მისი გამოთვლისათვის საჭიროა მხოლოდ დაკავშირების ატრიბუტების მნიშვნელობების სიის გადაცემა მთლიანი დამოკიდებულების გადაცემის მაგივრად და (2) მას "შემცირების" ეფექტი გააჩნია, რადგანაც $semijoin(rel_1, A \text{ op } B, rel_2)$ -ის შედეგები ყოველთვის rel_1 -ის ქვესიმრავლეს წარმოადგენს, როდესაც სრული დაკავშირება ყველაზე ცუდ შემთხვევაში დეკარტულ ნამრავლს აწარმოებს. როგორც უკვე 4.2 თავში აღინიშნა, რელიაციური გამოთვლის ტერმინებში ნახევარ კავშირი შეესაბამება გამოსახულებას ორი ცვლადით, სადაც ერთი ცვლადი არსებობის კვანტორითაა დაკავშირებული.

5. წვდომის გეგმები

წინა თავში განხილულ იქნა მოთხოვნის კომპონენტების ეფექტური გამოთვლის მეთოდები, რომელთა გამოყენებაც მოთხოვნის შესრულების ზოგადი ალგორითმის ფორმირებისათვის სამშენებლო ბლოკების სახით შეიძლება. მოთხოვნის შესრულების კარკასის ჩამოყალიბების საბოლოო ეტაპისათვის საჭიროა ამ ბლოკების გაერთიანება, დამოუკიდებელ სტანდარტიზირებულ, გამარტივებულ და გაუმჯობესებულ გამოსახულების ეფექტურ შესრულების პროცედურაში.

ასეთ პროცედურას შესასვლელში მიეწოდება ლოგიკურად დამუშავებული მოთხოვნა (როგორც მესამე თავში აღიწერებოდა), არსებული მონაცემების შენახვის სტრუქტურა, წვდომის

გზები და ღირებულების მოდელი. გამოსასვლელში ჩნდება ოპტიმალური (ან შემდგომისდაგვარად ევრისტიკულად "კარგი") წვდომის გეგმა. პროცედურა შემდეგი ნაბიჯებისაგან შედგება:

1. მოთხოვნის შესასრულებლად ყველა გონივრული ლოგიკური წვდომის გეგმის გენერირება. ლოგიკური წვდომის გეგმა აღწერს ოპერაციების მიმდევრობას ან შუალედური რეზულტატების მიმდევრობას რომელთაც არსებული დამოკიდებულებებიდან მოთხოვნის საბოლოო შედეგამდე მივყავართ.
2. ლოგიკური წვდომის გეგმების შევსება მონაცემების ფიზიკური წარმოდგენის დეტალებით (სორტირების მიმდევრობა, წვდომის ფიზიკური საშუალებების არსებობა, სტატისტიკური ინფორმაცია).
3. ყველაზე იაფი გეგმის არჩევა წვდომის და დამუშავების ღირებულების მოდელების გამოყენებით.

ამ თავში განიხილება წვდომის გზების გენერაცია, მათი ღირებულების შეფასების მოდელები და ყველაზე იაფი გეგმის შერჩევის პრობლემა. საბოლოო გეგმის ხარისხზე დიდ გავლენას ახდენენ არსებული შენახვის სტრუქტურები და წვდომის გზები, რომელთა ოპტიმიზაციაც ერთი მოულოდნელი მოთხოვნისათვის ჩვეულებრივ შეუძლებელია. ამიტომც 5.4 თავში მოკლედ განიხილება რამოდენიმე მოთხოვნის ერთდროული ოპტიმიზაცია.

5.1 წვდომის გეგმების გენერაცია

წვდომის გეგმები აღწერენ ოპერაციების (ოპერაციების გრაფების სახით წარმოდგენილი) ან შუალედური შედეგების (ობიექტის გრაფები) თანმიმდევრობას, რომელთაც არსებული დამოკიდებულებებიდან მოთხოვნის საბოლოო შედეგამდე მივყავართ. მოთხოვნის ოპტიმიზატორმა უნდა დააგენერიროს გეგმების საკმაოდ რაოდენობა, რათა მასში არსებობდეს ოპტიმალური გეგმა და საკმაოდ ზომიერი რათა ზედმეტი ხარჯი მისაღებ დონეზე შეინარჩუნოს.

განვიხილოთ [Smith and Chang 1975 и Yao 1979]-ს მიერ დემონსტრირებული ორი მიდგომა. სმიტი და ჩანგი მოთხოვნის გარდაქმნის "ავტომატურად პროგრამირებად" წესების ხისტ ნაკრებს იყენებენ (მესამე თავში განხილული წესების მსგავს). მათი პროცედურა აგენერირებს მხოლოდ ერთ გეგმას და არა აუცილებლად ოპტიმალურს. იაოს (Yao) მეთოდი აგენერირებს ყველა არა დომინირებად წვდომის გეგმებს რომელიც მოცემულ ფიზიკურ გარემოშია შესაძლებელი. თუმცა,

ეს მიდგომა შესაძლოა ორცვლადიანი მოთხოვნის კონტექსტში ვარგისი იყოს, ის მიუღებლად ძვირადღირებული ხდება უფრო რთული მოთხოვნებისათვის.

დანარჩენ მიდგომებში ხდება კომპრომისის მოძებნა ევრისტიკულ არჩევანსა და ალტერნატიული წვდომის გეგმების დეტალურ გენერაციას შორის. მაგალითად, System R [Chamberlin et al. 1981; Selinger et al. 1979]-ში გამოიყენება ევრისტიკული პროცედურა, რომელიც ეფუძნება SQL-ის ჩადგმულ ბლოკების კონცეფციას. უფრო დაბალ დონეზე ყოველი მოთხოვნის ბლოკისათვის გენერირდებიან და დარდებიან შესრულების გეგმები. მაღალ დონეზე მოთხოვნის ბლოკების შესრულების მიმდევრობა განისაზღვრება. [Kim 1982]-ში აღინიშნება, რომ მოთხოვნის ბლოკურ სტრუქტურას რომელიც მომხმარებელზეა დამოკიდებული ზედმეტი ყურადღება ექცევა და SQL-ის დამუშავებაში სტანდარტიზაციის ნაბიჯების შემოტანაა შემოთავაზებული.

5.2 წვდომის გეგმების ღირებულების ანალიზი

ფიზიკური წვდომის გეგმების არჩევანი განისაზღვრება ევრისტიკული წესებით ან ეფუძნება შენახვის სტრუქტურის და წვდომის ოპერაციის ღირებულების შეფასების მოდელს [Merrett 1977]. ამ თავში განიხილება ღირებულების შეფასების მოდელები და მათი ინტეგრაცია ოპტიმიზაციის პროცედურებში.

მიუხედავად იმისა, რომ რამოდენიმე მკვლევარი აანალიზებს მოთხოვნებს სამუშაო მეხსიერების [Kim 1982; Lang et al. 1977; Sacco and Schkolnick 1982] ან ცენტრალური პროცესორის ხარჯების მიმართ [Gotlieb 1975; Selinger et al. 1979], უმეტესი ღირებულების შეფასების მოდელები ეფუძნება მეორად მეხსიერებასთან მიმართვის რაოდენობას. მოცემული ოპერაციისათვის ამ სიდიდეზე გავლენას ახდენს ოპერანდების სიდიდე, გამოყენებული წვდომის სტრუქტურები და ძირითადი მეხსიერების ბუფერების ზომა.

გამოთვლის დასაწყისში ოპერანდებს წარმოადგენს მონაცემების არსებული სტრუქტურები რომელთა ზომაც ცნობილია, როგორებიცაა დამოკიდებულება და ინდექსი. თუმცა, უფრო გვიან სტადიებზე ოპერანდების უმეტესობა წარმოადგენს წინა ოპერაციების შედეგს და ღირებულების შეფასების მოდელმა უნდა შეაფასოს მათი ზომა მონაცემების საწყისი სტრუქტურის ინფორმაციისა და მათზე ჩატარებული ოპერაციების სელექციურობის გამოყენებით. თუმცა, ბევრი სამუშაო ტარდება მაგრამ, ჯერ შუალედური შედეგების შეფასების საყოველთაოდ აღიარებული ფორმულა არ არსებობს. ნაწილობრივ ეს არის შედეგი იმისა, რომ ბოლომდე არ

არის გაგებული, გამოყენებული ინფორმაციის დასაშვები მოცულობის თანაფარდობა შედეგის სიზუსტესთან.

უკანასკნელი წლების განმავლობაში მკვლევარები აცნობიერებენ მოთხოვნილებას მონაცემთა ბაზების მახასიათებლების შესახებ ყველა გამოყენებადი მოსაზრების ზუსტ ფორმულირებასა და კრიტიკულ ანალიზში, რათა მოხდეს ფორმალურად დასაბუთებული სისტემის პარამეტრების გენერაცია, რომელიც საშუალებას იძლევა:

1. მოხდეს ყველა შესაძლო ოპერაციისათვის შედეგის შეფასების გამოთვლა და
2. შუალედური შედეგების პარამეტრების მნიშვნელობების გამოთვლა შემდგომი ოპერაციებისათვის

საბოლოო ღირებულების შეფასებას წარმოადგენს მეორად მეხსიერებასთან მიმართების რაოდენობა და არა შუალედური შედეგების ზომები. კვლევების უმეტეს შემთხვევაში განიხილებოდა ამ ორ სიდიდეს შორის კავშირი [Cardenas 1975; Chan and Niamir 1982; Cheung 1982b; Luk 1983; Whang et al. 1983; Yao 1977a; Yu et al. 1978]. არსებითად, ის დამოკიდებულია გამოყენებულ შენახვის ფიზიკურ სტრუქტურასა და იმ ელემენტების პროპორციაზე რომელთაც ხდება მიმართვა.

დასასრულს აღვნიშნოთ, რომ ტრადიციულ მოსაზრებებს ატრიბუტების განაწილებასა და ელემენტების განთავსებაზე მივყევრთ ღირებულების გაზრდამდე და ამიტომაც ღირებულების შეფასებები ხელს უშლის პირდაპირი წვდომის სტრუქტურების გამოყენებას. მეორე მხრივ, უფრო მეტად სრულყოფილი მეთოდებისათვის მოითხოვება მონაცემთა ბაზაზე მეტი სტატისტიკური ინფორმაციის არსებობა. ასეთი ინფორმაციის შენახვის საკითხი დღემდე მთლიანად არ არის აღმოფხვრილი.

5.3 წვდომის გზების არჩევა

როგორ გამოიყენება შეფასების ღირებულება მოთხოვნების ოპტიმიზაციისას? როგორც 5.2 თავში აღინიშნებოდა, არსებობს ევრისტიკული პროცედურები რომლებშიც ისინი საერთოდ არ გამოიყენება. სხვა მიდგომებში შერჩევის ევრისტიკული შემცირება ერთიანდება ჩამოთვლით (enumerative) მინიმიზაციის ღირებულებასთან "თამაშის ბოლოს" [Youssefi and Wong 1979]. ექსპერიმენტები აჩვენებს, რომ მონაცემთა ბაზების ეფექტურობა შესაძლოა არსებითად გაზარდოს კომბინატორულმა ანალიზმა [Epstein and Stonebraker 1980].

შერჩევის პროცესში ღირებულების შეფასების გამოყენების ორი საშუალება არსებობს. პირველ რიგში, ყოველი წვდომის ალტერნატიული გეგმის ღირებულების სრულად განსაზღვრა შეიძლება [Blasgen and Eswaran 1976; Yao 1979]. ამ მეთოდის უპირატესობას წარმოადგენს პარალელიზმისა და უკუ კავშირის მსგავსი მეთოდების გამოყენების რეალისტური საშუალება. თუმცა, ამ შემთხვევაში ხარჯები ოპტიმიზაციაზე საკმაოდ მაღალია.

მეორე მხრივ, სტრატეგიის ღირებულება შეიძლება გამოვითვალოთ ინკრემენტულად მათი გენერაციის პარალელურად. ეს მიდგომა საშუალებას იძლევა მოვახდინოთ სტრატეგიების მთელი ოჯახის საერთო კომპონენტებით პარალელური შეფასება, რაც ოპტიმიზაციაზე ხარჯებს არსებითად ამცირებს. მაგალითად, [Rosenthal and Reiner 1982]-ში შემოთავაზებულ მეთოდში ინახება შუალედური შედეგის მიღების მხოლოდ მინიმალური საშუალების ღირებულება და უარყოფილია ყველა სხვა მეთოდი, რომელიც არაოპტიმალურად არის მიჩნეული.

ამ მეორე მეთოდის გაფართოებას წარმოადგენს მოთხოვნების ოპტიმიზაციის დინამიური პროცედურა, რომელიც იმ დაკვირვებიდან გამომდინარეობს, რომ დროის ყოველ მომენტში საჭიროა გადაწყვეტილების მიღება მხოლოდ მომდევნო ოპერაციის შესრულების თაობაზე. იმისათვის, რომ საერთო ოპტიმალურობის გარანტირება მოხდეს, ამ გადაწყვეტილების მხოლოდ შედეგები უნდა შეფასდეს გამოთვლების დარჩენილი ნაწილისათვის. დინამიურ პროცედურებს გააჩნია რეალური მონაცემები მის ყველა ოპერანდზე, შუალედური შედეგების ჩათვლით. ეს ინფორმაცია შესაძლოა ასევე გამოყენებულ იქნას დარჩენილი ნაბიჯების შეფასებების გასაახლებლად. დინამიურ მეთოდს ორი ნაკლი აქვს: მისი ღირებულება და ლოკალურ სივრცეში შეცდომის საფრთხე. ამ მეთოდის აკურატულად გამოყენებისას შესაძლებელია იმ მოთხოვნების შეფასების ღირებულების შემცირება, რომლებშიც რეალური შუალედური მონაცემების ზომები განსხვავდება შეფასებული ზომებისაგან.

5.4 რამოდენიმე მოთხოვნის მხარდაჭერა

აქამდე განხილული მოთხოვნების გამოთვლის ყველა პროცედურა კონცენტრირებული იყო თითო მოთხოვნის შესრულების ოპტიმიზაციაზე. [Chesnais et al. 1983] ასევე გამოიკვლევა მონაცემთა ბაზასთან რამოდენიმე მომხმარებლის პარალელური წვდომის საკითხზე გავლენა. თუმცა, მოთხოვნის ოპტიმიზაციის სტრატეგიები შესაძლოა ასევე უბრალო პარალელიზმის ფარგლებს ცდებოდეს საერთო ოპერაციების შესრულების ღირებულების გაზიარებით (sharing) რამოდენიმე მოთხოვნას შორის. გარდა ამისა, სტრატეგია, რომელიც რამოდენიმე მოთხოვნის

ერთდროულ შესრულების ოპტიმიზაციას ახდენს, შესაძლოა რომ ითვალისწინებდეს ინვესტიციებს დამატებით წვდომის გზებში, რომელთა შექმნაც არ იქნებოდა ხელსაყრელი თითო მოთხოვნის ღირებულების მიზეზით. რამოდენიმე მოთხოვნის ოპტიმიზაცია შეიძლება სამ ჯგუფად დაიყოს, იმისდა მიხედვით თუ როდის მიიღება გადაწყვეტილებები:

1. პაკეტური მოთხოვნების ერთდროული ოპტიმიზაცია,
2. ინდექსების შერჩევა და
3. მონაცემთა ბაზის ფიზიკური პროექტირება.

პაკეტური დამუშავების მეთოდი ანალოგიურია 4.3 თავში აღწერილი მეთოდებისა, რომლებიც გამოიყენებოდა მრავალცვლადიანი გამოსახულების გამოსათვლელად. პირველ რიგში მოთხოვნის გამოთვლისას შეიძლება გამოიყენებოდეს საერთო ქვე-გამოსახულებების შედეგები და თუ განსხვავებული ქვე-გამოსახულებების გამოთვლისას საჭიროა ერთიდაიმავე მონაცემთა ფიზიკურ გვერდთან მიმართვა, მაშინ ამის გაკეთება შესაძლებელია მეორად მეხსიერებასთან ერთი მიმართვით. მეორე მხრივ, შეიძლება უზრუნველყოფილ იქნეს დროებითი წვდომის ფიზიკური გზები, როგორებიცაა სორტირება, ქეშირება ან ინდექსები, რომელთა ღირებულება მთლიანად პაკეტისათვის ანაზღაურდება.

ინდექსების გამოყენების მნიშვნელობა მოთხოვნის გამოთვლის ალგორითმების ეფექტურობაში ხშირად აღინიშნებოდა, ამ თვალსაზრისით ინდექსები იშვიათად აყენებენ ზიანს, თუმცა მათი გამოყენება ყველაზე მეტად სასარგებლოა თუ ძალიან სელექტიურები არიან და მოთხოვნაში ხშირად გამოყენებად ატრიბუტებთან წვდომას უზრუნველყოფენ. თუმცა ინდექსის შერჩევისას აუცილებელია გათვალისწინებულ იქნას ცვალებდი ტრანზაქციები, რადგანაც საბაზისო მონაცემების ცვლილებასთან ერთად მათ ინდექსების ცვლილებაც უნდა მოახდინონ.

მოთხოვნის ოპტიმიზაცია მონაცემთა ბაზის ფიზიკურ პროექტირებაზე ახდენს გავლენას. თუმცა, გრძელვადიანი ოპტიმიზაცია წარმოადგენს მონაცემთა ბაზის ფიზიკური პროექტირების მხოლოდ ერთ ასპექტს. მნიშვნელოვანი საპროექტო სტრატეგიები, რომლებიც მოთხოვნის დამუშავების ეფექტურობაზე ახდენენ გავლენას შეიცავენ დამოკიდებულებების ატრიბუტების მნიშვნელობების მიხედვით ელემენტების ჰორიზონტალურ კლასტერიზაციას და ატრიბუტების ვერტიკალურ დაყოფას საერთო მიმართვების სიხშირის მიხედვით.

დასკვნა

ნაშრომში შესრულდა მონაცემთა ბაზებში მოთხოვნების ლოგიკური გარდაქმნებისა და ფიზიკური შესრულების მეთოდების განხილვა რელიაციური გამოთვლის კარკასის გამოყენებით. ნაჩვენები იქნა, რომ მოთხოვნის ეფექტური დამუშავების პრობლემის გადასაწყვეტად ტრადიციულ ცენტრალიზირებულ და დანაწილებულ მონაცემთა ბაზის სისტემებში დიდი ცოდნაა დაგროვებული.

ოპტიმიზაციის კვლევების სამომავლო მიმართულებები მოიცავენ ღირებულების მართვას მაგრამ რეალური შეფასებების შემუშავებას, მოთხოვნების ოპტიმიზაციას დედუქტიური ან გამოთვლითი შესაძლებლობებით და რამოდენიმე მოთხოვნისა და განახლების ტრანზაქციის ერთდროულ ოპტიმიზაციას.

მიუხედავად, მოთხოვნების ოპტიმიზაციის შემუშავებული მეთოდების დიდი რაოდენობისა ეს ამოცანა, კვლავ აქტუალური რჩება არსებული მონაცემთა ბაზების მართვის სისტემებისათვის. როგორც, ამ სფეროში ცნობილი სპეციალისტი ს.ჩაუდხარი აღნიშნავს: «მოთხოვნების ოპტიმიზაცია, წარსულთან შედარებით, კვლავინდებურად თუ უფრო მეტადაც კი აქტუალურია. თანამედროვე მონაცემთა ბაზები შეიცავენ რეალურ დროში ტრანზაქციების დამუშავების სისტემას, ERP-სისტემას, კლიენტთან ურთიერთობის მართვის სისტემებს, რეალურ დროში ანალიტიკურ დამუშავებას, მონაცემთა ანალიზს მონაცემთა საცავების გამოყენებით (Data-warehouses). ამ დანართებით გენერირებული მოთხოვნები სულ უფრო და უფრო რთულია, ხოლო მონაცემთა ბაზები -კი, ისეთი დიდი როგორც არასდროს. ამგვარად, მოთხოვნების ოპტიმიზაციას, რომელიც ეძებს დამუშავების განსხვავებულ სტრატეგიებს და არჩევს შესრულების საუკეთესო გეგმას, კვლავინდებურად ცენტრალური როლი აკისრია»

ლიტერატურა

1. AHO, A. V., BEERI, C., AND ULLMAN, J. D. 1979a. The theory of joins in relational databases. Database Syst. 4, 3 , 297-314.
2. ROSENTHAL, A., AND REINER, D. 1984. Querying relational views of networks. In Query Processing in Database Systems, W. Kim, D. Reiner, and D. Batory.
3. SHULTZ, R. g., AND ZINGG, R. J. 1984. Response time analysis of multiprocessor computers for database support. 100-132.
4. SMITH, J. M., AND CHANG, P. Y. W. 1975. Optimizing the performance of a relational algebra database interface, 568-579.
5. YAO, S. B. 1979. Optimization of query evaluation algorithms. ACM Trans. Database Syst. 4, 2 , 133-155.
6. http://www.madgik.di.uoa.gr/sites/default/files/acm_tods_v31.2.pp537-583.pdf
7. http://citforum.ru/database/articles/art_26.shtml
8. <http://www.sdteam.com/t1951>
9. <http://www-i5.informatik.rwth-aachen.de/lehrstuhl/lehre/IDB05/download/Literature/QO.pdf>
10. http://www.ispras.ru/ru/proceedings/docs/2012/23/isp_23_2012_195.pdf